Simulation software for safe, sustainable and future deltas

Delft3D FM Suite

Deltares systems

D-Real Time Control

User Manual

Deltares

D-Real Time Control

User Manual

Released for:
Delft3D FM Suite 2023
D-HYDRO Suite 2023
SOBEK Suite 3.7

Version: 1.4
Revision: 78699

25 April 2024

D-Real Time Control, User Manual

Published and printed by:
Deltares
Boussinesqweg 1
2629 HV Delft
P.O. 177
2600 MH Delft
The Netherlands

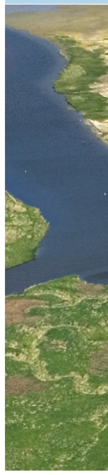
telephone: +31 88 335 82 73
e-mail: Information
www: Deltares

For sales contact:
telephone: +31 88 335 81 88
e-mail: Sales
www: Sales & Support

For support contact:
telephone: +31 88 335 81 00
e-mail: Support
www: Sales & Support

Contents

List of Tables



List of Figures

1 A guide to this manual

1.1 Introduction

This User Manual concerns the module D-Real Time Control.

This module is part of several Modelling suites, released by Deltares as Deltares Systems or Dutch Delta Systems. These modelling suites are build with use of the Delta Shell framework. The framework enables to develop a range of modeling suites, each distinguished by the components and — most significantly — the (numerical) modules, which are plugged in. The modules which are compliant with the Delta Shell framework are released as D-*Name of the module*, for example: D-Flow Flexible Mesh, D-Waves, D-Water Quality, D-Real Time Control and D-Rainfall Runoff.

Therefore, this User Manual is shipped with several modelling suites. On the *Start Page* links are provided to all relevant User Manuals (and Technical Reference Manuals) for that modelling suite. Other user manuals can be referenced. In that case, you need to open the specific user manual from the *Start Page* in the central window. Some texts are shared in different user manuals, in order to improve the readability.

1.2 Overview

To make this manual more accessible we will briefly describe the contents of each chapter.

If this is your first time to start working with D-RTC we suggest you to read chapter 3, Getting started/Tutorial. This chapter provides a tutorial.

Chapter 2: Overview gives a brief introduction on D-RTC.

Chapter 4: All about the modelling process provides practical information on the GUI, setting up so-called *Control groups* presented as a Flow chart and validating the model.

Chapter 5: Simulation and model output describes how the simulation results can be accessed.

Chapter 6: Technical reference gives technical background information on the principles of feedback control.

1.3 Manual version and revisions

This manual applies to SOBEK 3 suite (version 3.7 and higher), D-HYDRO Suite (version 2018 and higher) and Delft3D Flexible Mesh Suite (version 2018 and higher).

1.4 Changes with respect to previous versions

New in this edition is Chapter 6: Technical reference.

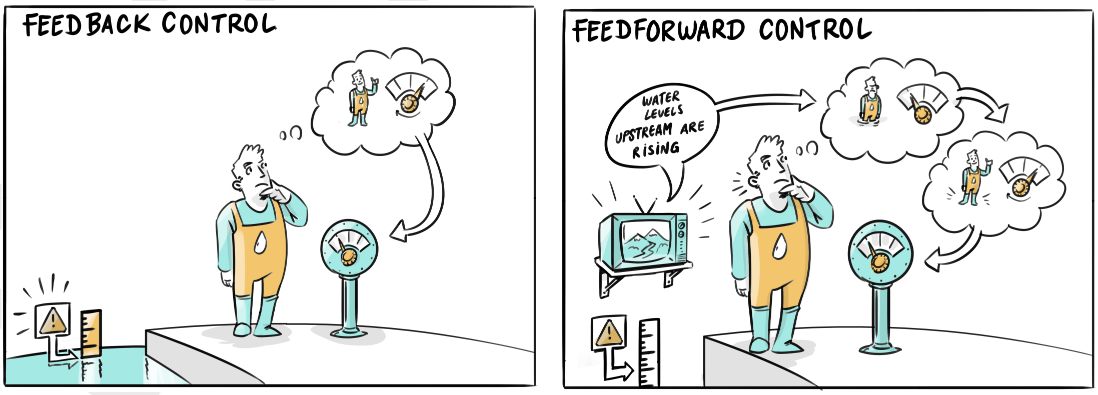
2 Overview

The D-RTC (Real Time Control) plug-in can be used for the modelling of feedback control of hydraulic structures. It can be applied to rainfall-runoff, hydraulics and water quality computations. The D-RTC module is used in a integrated model and is always combined to a hydrodynamic model, such as D-Flow 1D or D-Flow FM.

2.1 Feedback control and feedforward control

Feedback is a control principle where the control actions are determined based on a control error (i.e. the difference between target value and actual value). The control action feeds back on the control error. Feedforward control uses a control signal from outside the system to determine control actions. The control action does not impact the feedforward control signal.

Figure 2.1 shows an example of these techniques. In both cases, the operator aims to maintain the water level below a certain threshold by means of a hydraulic structure for flood protection. In Figure 2.1a the operator controls the water level with the help of information from the system: if the water level reaches a certain level, he takes action, and the action feeds back on the water level itself. When measurements of disturbances are used to form control decisions the control method is called feedforward control. In Figure 2.1b the operator checks information from a location which is outside the system (e. g. a weather forecast) and controls accordingly. This allows him to anticipate to certain extent on a future situation, and his control operation does not affect the information his decision is based on.



***(a)** feedback control* ***(b)** feedforward control*

***Figure 2.1:** Feedback control and feedforward control*

Feedback control and feedforward control can usually be represented as a flow chart or decision tree with the following elements:

- ◇ trigger (condition)
- ◇ operating rules (controller-actuator)
- ◇ input data location (connection to an observation point or a measurement station, e.g. a river gauge)
- ◇ output data location (connection to a structure node)

An example of such a flow chart is given in Figure 2.2.

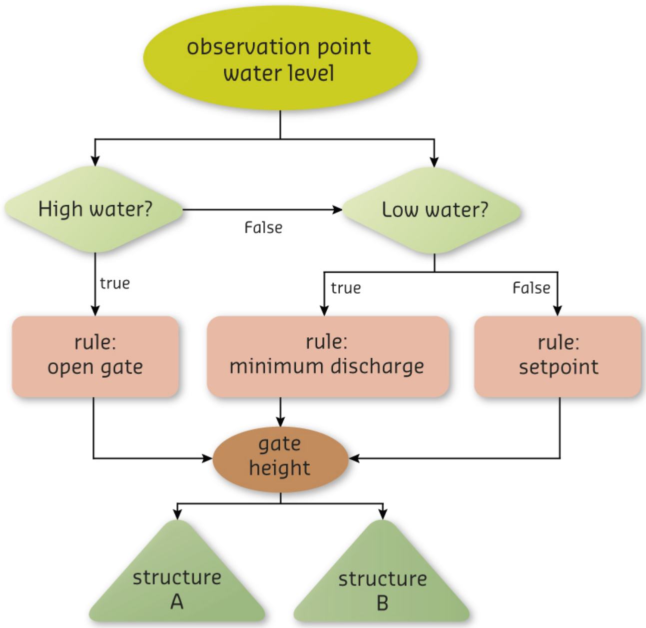


***Figure 2.2:** Example flow chart with feedback control*

A trigger implements conditions for

◇ defining when an operating rule / controller or another trigger is applied
◇ returning true or false, e.g. if a threshold is crossed or not.

A trigger represents a switch, an alarm level or a man-made decision in a real-time control model.

Usually it returns true and false, but there are also trigger implementations that return numierical values.

An operating rule

◇ defines how a structure operates, and
◇ returns a value for a controlled parameter, e.g. a gate opening or turbine release.

Very simple operating rules define operational modes for hydraulic structures like "pump switched on" and "pump switches off", or "gate open" and "gate closed".

In this case the rule does not specify how exactly the hydraulic structure is operated, but for many model applications this is sufficient.

More advanced operating rules comprise closing speeds for gates or model a complete controller actuator system. An example for such an operating rule is a motor (the actuator) drives the segment of a weir and is switched on and off by a floating switch (the controller itself).

2.2 D-RTC windows

A D-RTC model has three main windows in the GUI framework:

1 the **Project** window,
2 the **map** with the **Flow chart**, and

3 the **Properties** window.

The **Project** window is used to show a total overview of all D-RTC model objects, while the map and flow chart show the relations between the D-RTC components. Figure 2.3 shows an example of a D-RTC-model in the **Project** window. In this case the composite model contains a D-Flow 1D model and the D-RTC model. The D-RTC model consists of a set of controlgroups and an output folder. This is described in more detail in section 2.3. Figure 2.4 shows an example of a flowchart. Flowcharts are described in section 2.4. The **Properties** window shows details of the RTC-components and facilitates editing (see also section 2.5). Figure 2.5 shows an example of the **Properties** window for a Time rule (see also chapter 4).

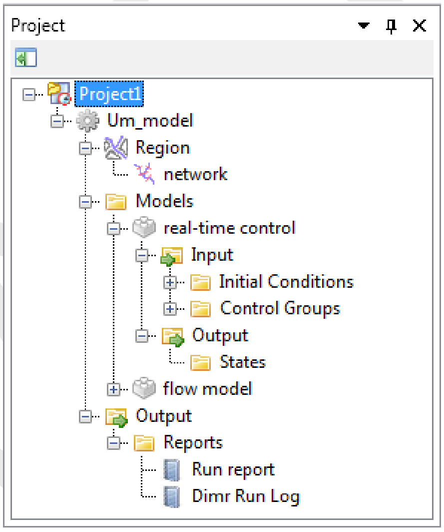


Figure 2.3: *Example of an RTC-model in the **Project** window*

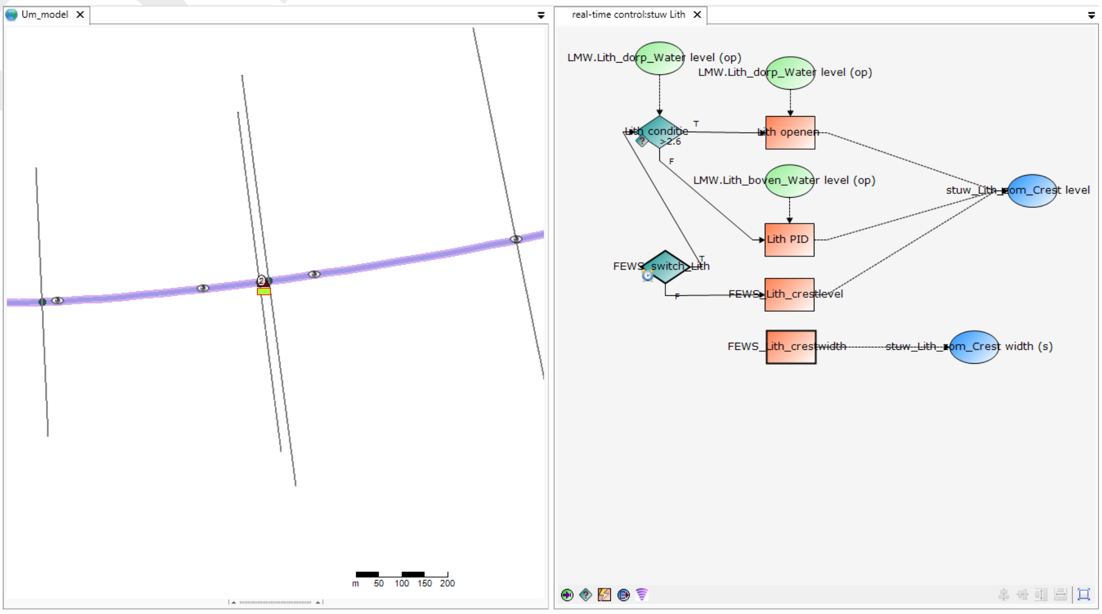


Figure 2.4: *Example of a flow chart*

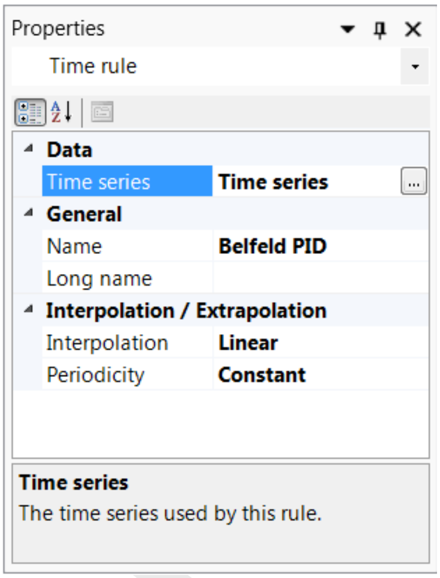


Figure 2.5: *Example of the properties window for a Time rule*

Figure 2.6 gives an overview of the RTC modelling concept. D-RTC uses values from a connected hydro model as control parameters to decide on control actions and to determine the values of controlled parameters. The values of controlled parameters are fed back to the hydro model. Examples for control parameters are:

- ◇ Hydraulic parameters at an observation point, such as discharge or waterlevel
- ◇ Water quality parameters at an observation point
- ◇ Rainfall
- ◇ External data, such as meteorological conditions, diversions due to building or maintenance activities etc.

Examples for controlled parameters are:

- ◇ Crest level or crest width of weirs
- ◇ Discharge of pumps
- ◇ Gate opening at gated weirs
- ◇ Valve opening at Culverts

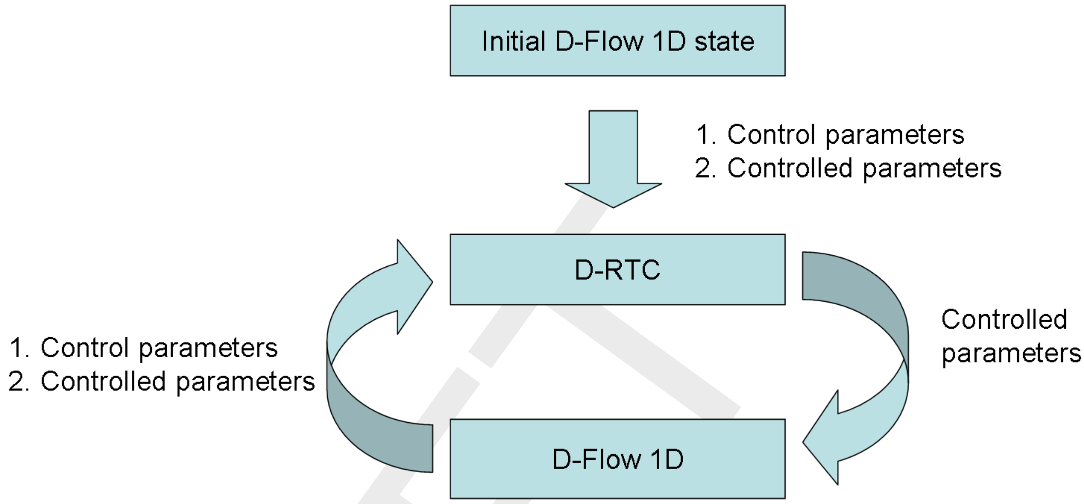


Figure 2.6: *D-RTC modelling concept and data flow*

D-RTC uses the control parameters (i. e. the data from a connected hydro model) to evaluate conditions. The conditions trigger the evaluation of rules, and within the rules the controlled parameters are determined. For example, if a pump operates with a switch-on level of 1.5 m and has a capacity of 500 $m^3 s^{-1}$, the condition "switch-on level reached" is true if the connected hydro model computes a value of 1.5 m or larger. The condition is false if the water level is lower than 1.5 m. The rule connected to the true-output sets the discharge to 500 $m^3 s^{-1}$, while the rule connected to false-output set the discharge of the pump to 0 $m^3 s^{-1}$.

- Rules contain the actual algorithms to calculate the values of a controlled parameter.
- Conditions trigger a rule. Both the true and false outcome of a condition can be used to activate rules.

A control flow is a sequence of connected conditions and rules, the control flow should end in a controlled parameter. D-RTC visualizes control flows as a flowchart, which is described in more detail below.

D-RTC uses the objects from a hydraulic model such as a D-Flow 1D model, but has no knowledge of the model itself and no spatial information. The objects from a hydraulic model used by D-RTC are passive objects which can not be edited in D-RTC. D-RTC directs the control flow, which means that the only editable objects are rules and conditions.

2.3 Controlgroup

A D-RTC model consists of one or multiple controlgroups which are shown in the **Project** window. A controlgroup is a set of D-RTC components. Each controlgroup consists of

- one flow chart with one ore more controlflows (explained in section 2.4)
- a list of observation points which supply the values of the control parameters
- a list of conditions used in the controlflow(s)
- a list of rules used in the controlflow(s)
- a list of structure output locations for the controlled parameters

The set of elements in a single flowchart is a single controlgroup and one or more controlgroups form a D-RTC model.

The controlgroup can be used to organize the D-RTC model. For example, a user can decide to group the controlflows per

- ◇ controlled parameter; each controlled parameter has its own controlflow and a controlgroup has one controlflow,
- ◇ controlled structure; a controlled structure can have controlflows for each controlled parameter (for example, both crest level and crest width are controlled of a single weir). Each controlgroup can then have more than one controlflow,
- ◇ compound structure; several structures combined can form one large complex, such as Haringvlietsluizen, which consist of 17 individual locks. It can be convenient to group D-RTC around these compound structures. Each controlgroup can then contain more than one controlflow, one for each controlled parameter.

Deciding how to group the controlflows is finding a balance between transparancy and easier modeling in the controlgroups, and having a good overview of the total model. It is recommended to use as few controlflows as possible within a single controlgroup. Only use more than one controlflow per controlgroup if the **Project** window becomes too complex.

2.4 Flow chart

The flow chart is the visual interpretation of the controlgroup. While the **Project** window only shows a list of all components of a controlgroup, the flowchart shows how the components of the controlgroup relate to one another.

D-RTC is built around the concept of controlflows. Figure 2.7 shows the concept of a controlflow and its components. The controlflow in a flow chart always:

- ◇ has one starting point for each controlled parameter (depicted with a thick black line around the controlflow component, see Figure 2.7),
- ◇ has at least one controlled parameter and one rule,
- ◇ can combine multiple conditions and rules,
- ◇ shows the controlflow with solid arrows, and data input or output with dashed arrows, see Figure 2.7,
- ◇ has exactly one active path per controlled parameter (no 2 rules for the same controlled parameter are active at the same time).

In section 2.6 examples are described in more detail.

2.5 The Properties window

Similarly to other suite modules in D-RTC the **Properties** window shows details of a D-RTC schematisation object by clicking on an item in the **Project** window or on the flowchart. The corresponding parameters can be edited in this window. An additional table window is shown if necessary. Examples for properties of different D-RTC objects are given below:

- ◇ Condition parameters:
  - ◻ condition type
  - ◻ the table in case of a lookup table controller
- ◇ control parameters:
  - ◻ discharge
  - ◻ waterlevel
  - ◻ salinity
- ◇ controlled locations and parameters

◇ Rule parameters:
  - rule type
  - the table of a lookup table controller
  - rule-specific parameters

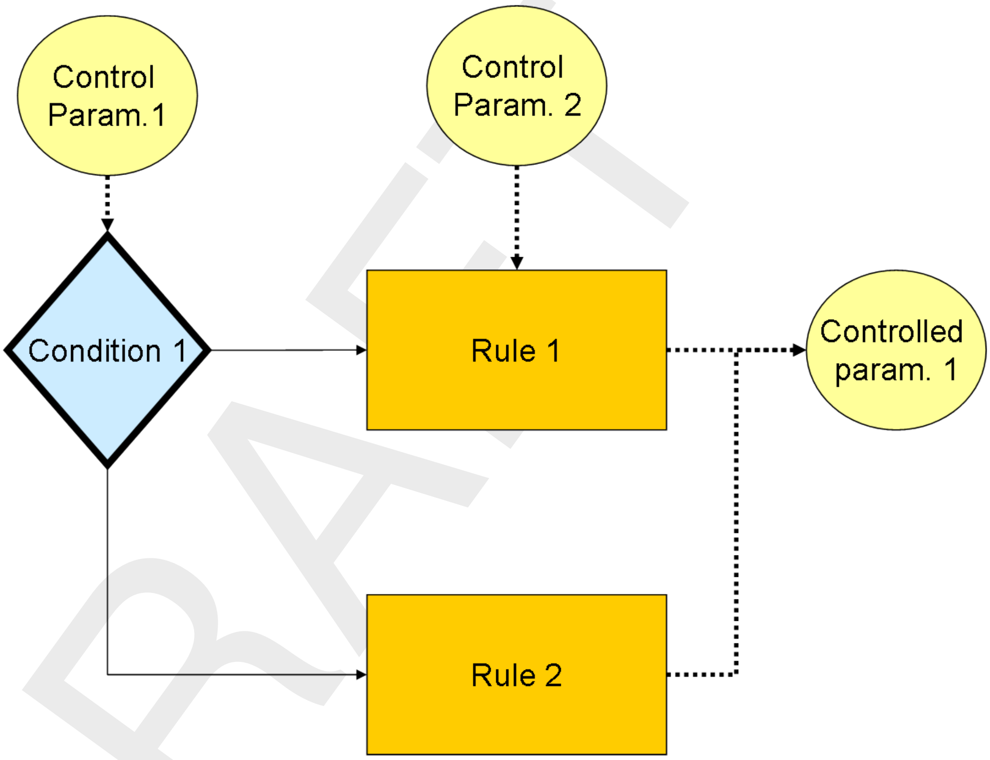


***Figure 2.7:** Components and basic concept of the flowchart*

2.6 Examples

In this section several examples of controlflows are presented. These examples also form the basis for the tutorial in chapter 3.

2.6.1 Minimal controlflow

Figure 2.8 shows the minimal controlflow; a parameter controlled by a rule. Depending on the type of rule, there may be also be control data required. In addition to a rule, conditions can be added that (de)activate a rule, Figure 2.9. An example for Figure 2.8 could be that the water level is 3 m on the first day and 6 m on day two at a specified location. A condition for that rule is added in Figure 2.9 and could imply that the rule is only active when the discharge at a certain observation point is higher than a specified value.

Chapter 4 explains which types of rules and conditions are available and what they do.

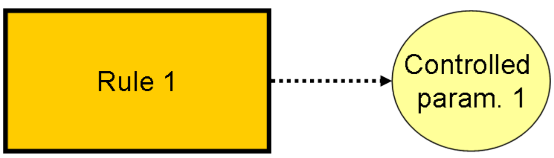


***Figure 2.8:** Example minimal controlflow*

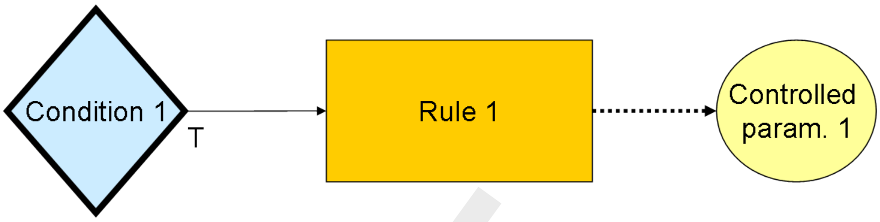


Figure 2.9: *Example minimal controlflow with a condition*

2.6.2 Combinations of conditions and rules

In this section an overview is given of basic combinations of conditions and rules.

A condition can be used on its own, but will often be used in combinations. The most elementary combinations of two conditions are AND ($\wedge$) and OR ($\vee$). An AND ($\wedge$) combination of two conditions means that both conditions have to be true for the rule to be active. As soon as one of the two conditions is false, the rule becomes inactive. The controlflow first checks the first condition, and only if this condition is true, the controlflow proceeds to check the second condition. If either the first or second condition is false, another rule can be activated. It is not required to have a different rule for the false-scenario; if a structure only has to perform an action if the conditions are true, no second rule is required. A second rule is only required if the structure also has to perform an action once the conditions are false.

An OR ($\vee$) combination of two conditions means that only one of the two conditions has to be true for the rule to be active. Only if both conditions are false, the rule becomes inactive.

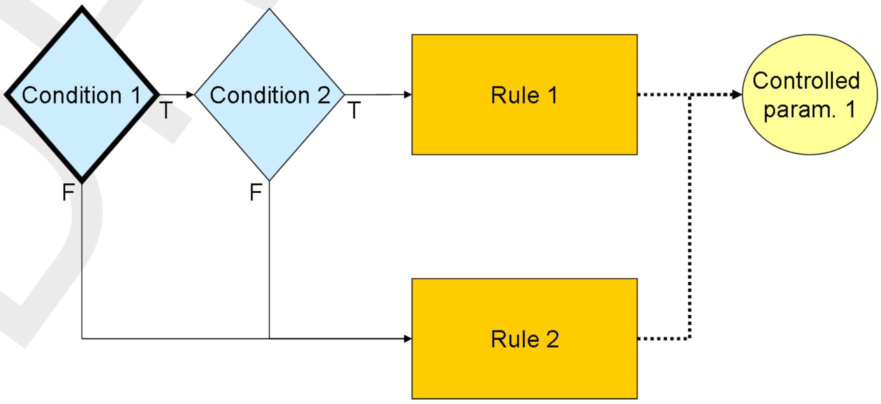


Figure 2.10: *Example of two conditions that form an AND ($\wedge$) condition.*

Controlflows as shown in Figure 2.10:

- Rule 1 is performed if (*condition 1* $\wedge$ *condition 2*) is TRUE.
- Rule 2 is performed if (*condition 1* $\vee$ *condition 2*) is FALSE.

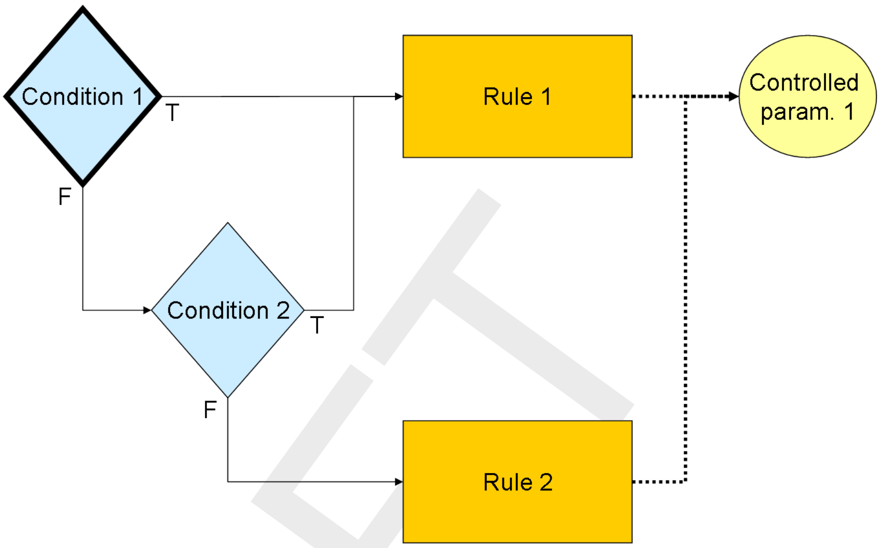


Figure 2.11: *Example of two conditions that form an OR (∨) condition.*

Controlflows as shown in Figure 2.11:

- Rule 1 is performed if (*condition 1* ∨ *condition 2*) is TRUE.
- Rule 2 is performed if (*condition 1* ∧ *condition 2*) is FALSE.

By adding conditions the controlflow may be expanded to more complex situations. Figures 2.12 and 2.13 show some possibilites by using three conditions in a single controlflow. All possibilities for combinations of three conditions are:

- $1 \wedge 2 \wedge 3$
- $1 \vee 2 \vee 3$
- $1 \wedge (2 \vee 3)$ (shown in Figure 2.12)
- $1 \vee (2 \wedge 3)$ (shown in Figure 2.13)

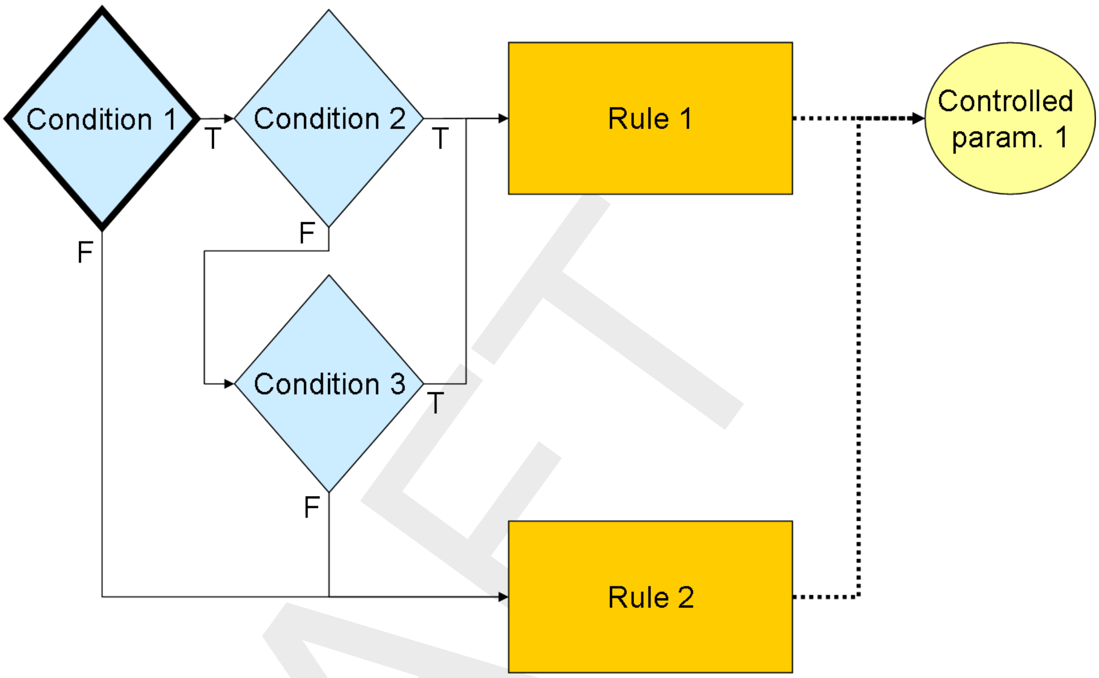


Figure 2.12: *Example of three conditions.*
Rule 1 is performed if (condition 1 ∧ (condition 2 ∨ condition 2)) is TRUE.

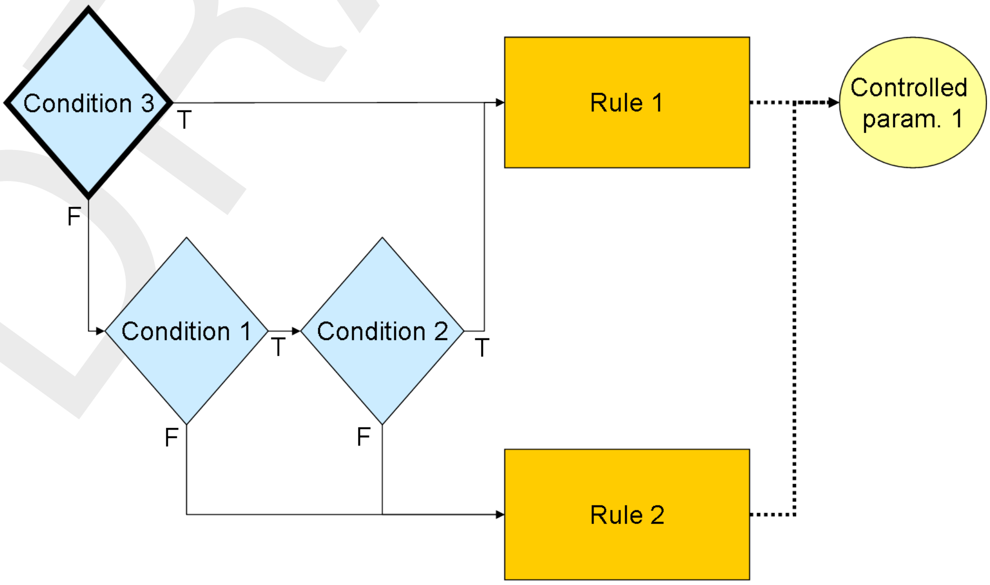


Figure 2.13: *Example of three conditions.*
Rule 1 is performed if (condition 1 ∨ (condition 2 ∧ condition 2)) is TRUE.

Similarly, the control flow can be extended with even more conditions and rules. Figures 2.14 to 2.19 give examples for a situation with four conditions.

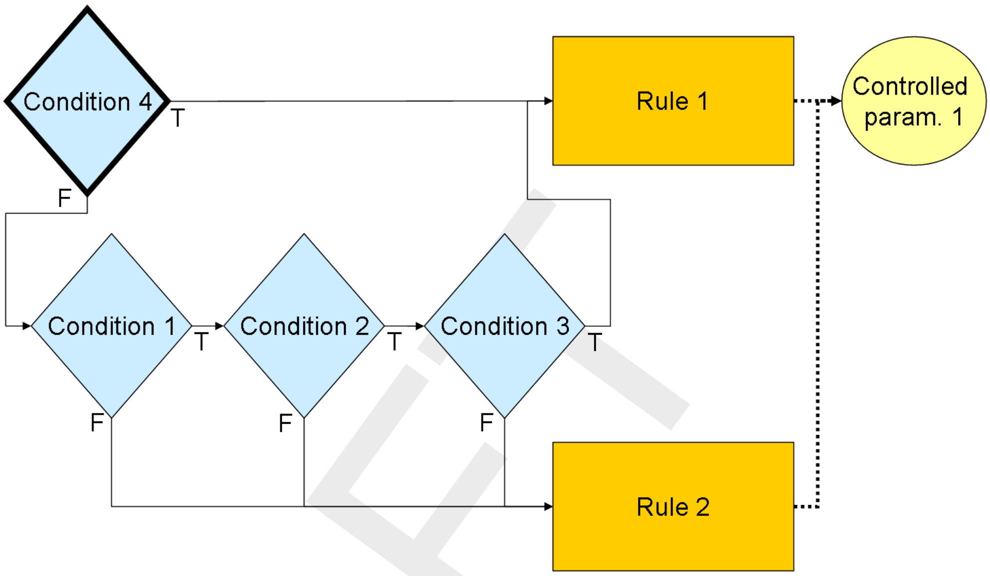


Figure 2.14: *Example of four conditions.*
Rule 1 is performed if (condition 4 $\vee$ (condition 1 $\wedge$ condition 2 $\wedge$ condition 3)) is TRUE.

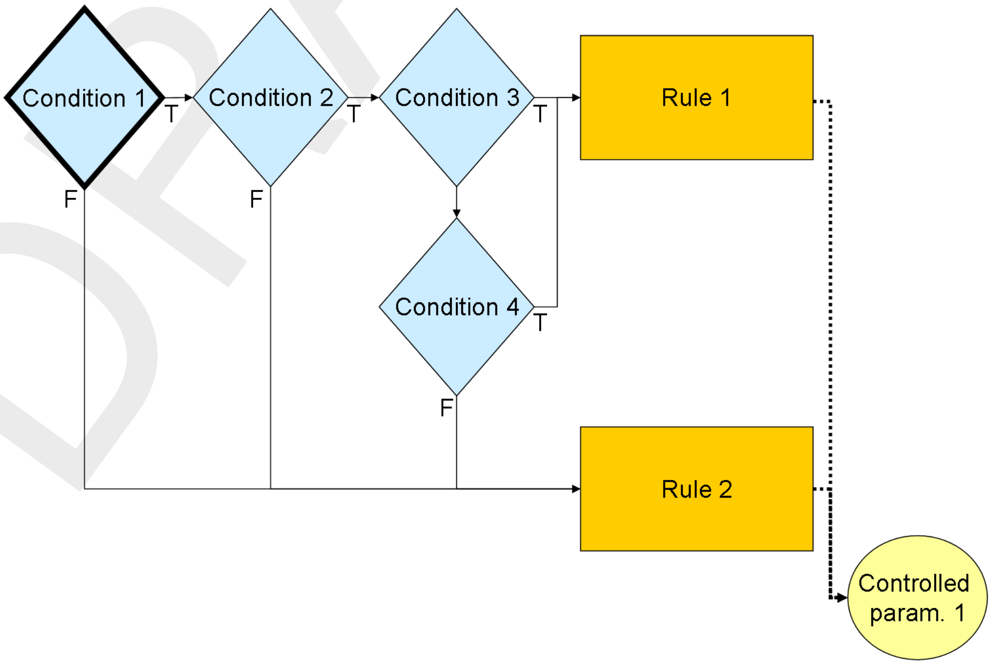


Figure 2.15: *Example of four conditions:*
Rule 1 is performed if (condition 1 $\wedge$ condition 2 $\wedge$ (condition 3 $\vee$ condition 4)) is TRUE.

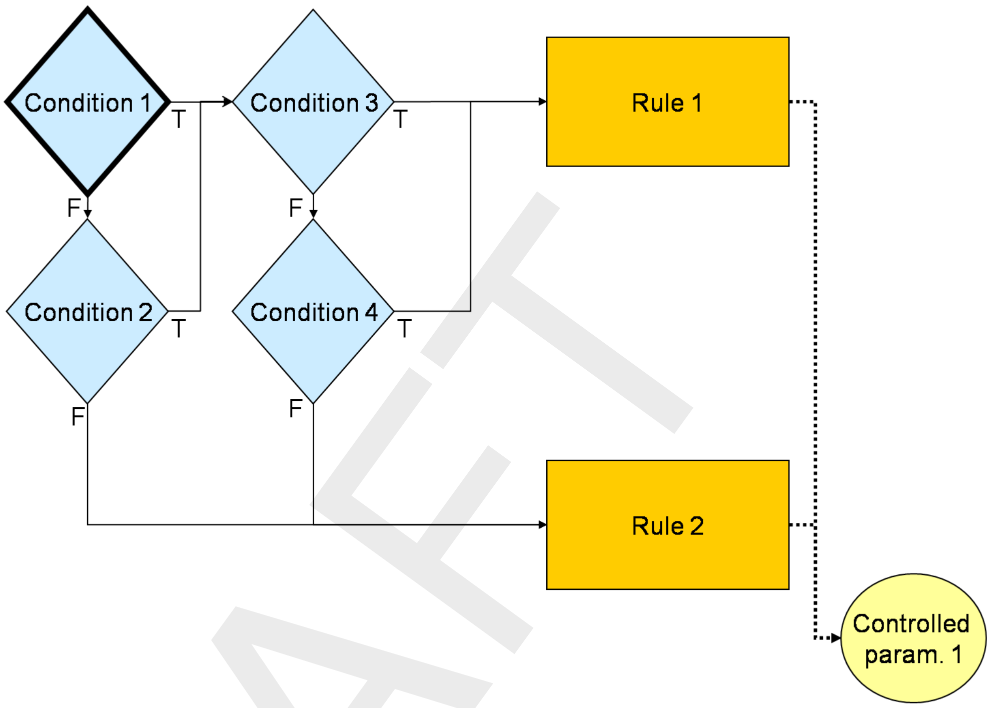


Figure 2.16: *Example of four conditions.*
Rule 1 is performed if (condition 1 ∨ condition 2) ∧ (condition 3 ∨ condition 4) is TRUE

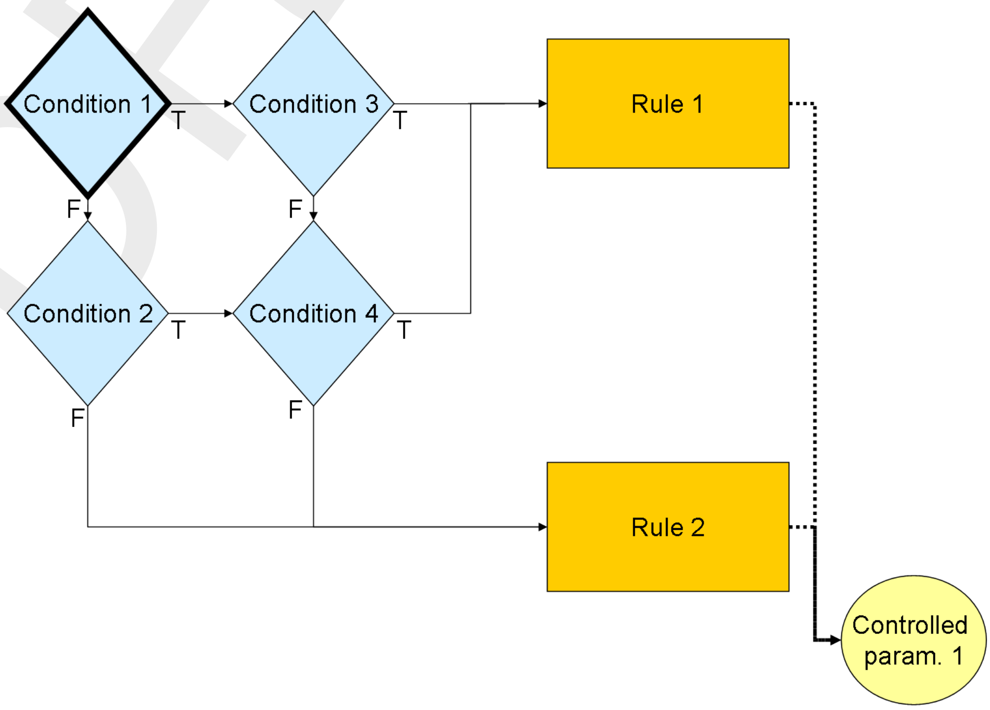


Figure 2.17: *Example of four conditions.*
Rule 1 is perfomred if (condition 1 ∧ (condition 3 ∨ condition 4) ∨ (condition 2 ∧ condition 4))

Note the difference between Figures 2.16 and 2.17; there is only a small difference in flowchart (the arrow going from the true side of Condition 2 to Condition 3), but a large difference in meaning!

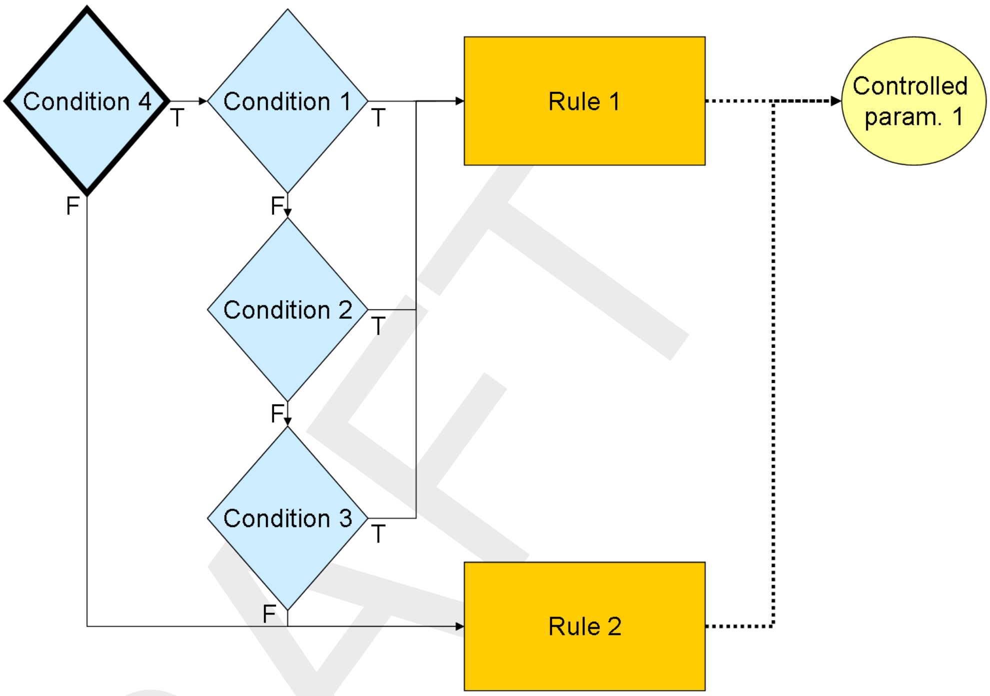


Figure 2.18: *Example of four conditions.*
Rule 1 is perfomred if (condition 4 ∧ (condition 1 ∨ condition 2 ∨ condition 3)) is TRUE

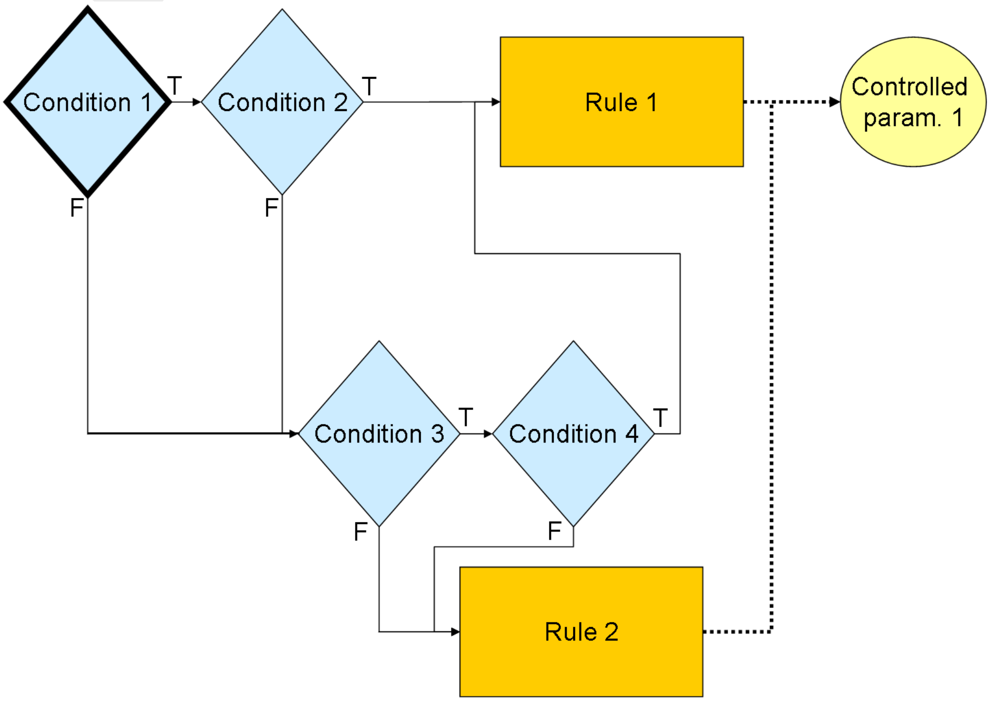


Figure 2.19: *Example of four conditions.*
Rule 1 is performed if ((condition 1 ∧ condition 2) ∨ (condition 3 ∧ condition 4)) is TRUE

3 Getting started/Tutorial

3.1 Introduction

In this chapter we will provide examples of D-RTC with a tutorial. The D-RTC module can not be used on its own: a D-RTC model uses information and determines the values of parameters of another model, typically a D-Flow 1D model or D-Flow FM model, such a combination is called a integrated model. However, this can also be a water quality model or a rainfall-runoff model. We start with the tutorial models for D-Flow 1D and D-Flow FM integrated with D-RTC.

3.2 The integrated model

Applying control with D-RTC on a D-Flow 1D or D-Flow FM model is integrated modeling (or coupled modeling). To set up an integrated model in a suite is similar for a D-Flow 1D model and a D-Flow FM model.

➢ Start the SOBEK Suite or Delft3D Flexible Mesh Suite,
➢ In the **Project** window, right-mouse click on the project, e.g. <Project1> and select *Add* → *New Model. . .*

A **Select model . . .** dialog appears, Figure 3.1.

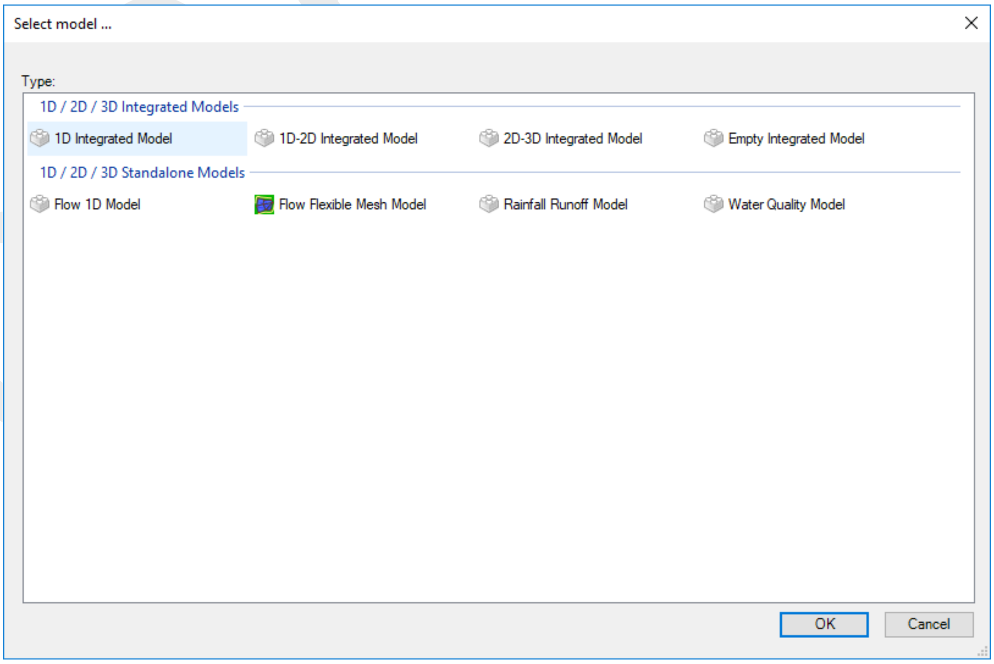
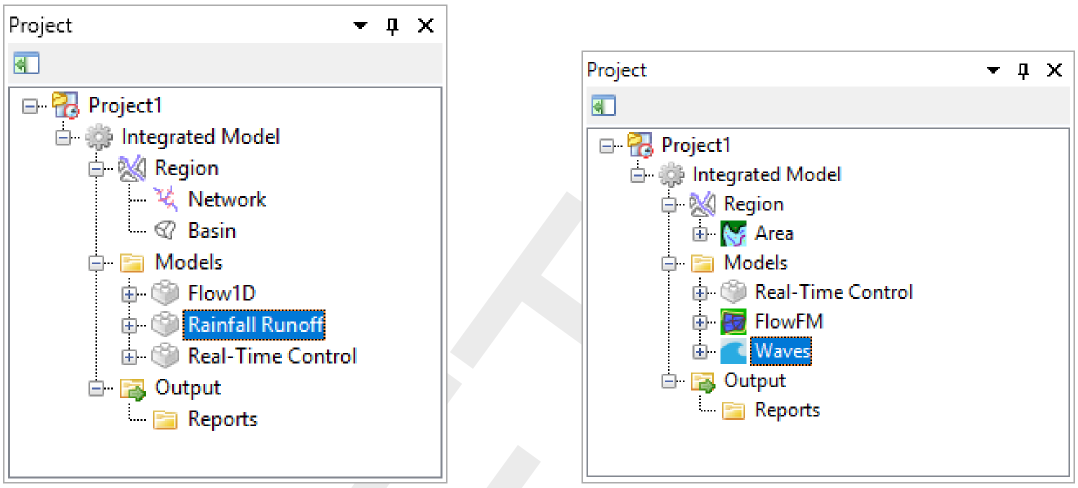


Figure 3.1: *Select a new model:* 1D Integrated Model *for a D-Flow 1D or* 2D-3D Integrated Model *for a D-Flow FM integrated model*

➢ Choose “1D Integrated Model” or “2D-3D Integrated Model”.

From the **Project** window remove the D-RR or D-Waves entry.



(a) Delete the indicated Rainfall-Runoff *model* *(b) Delete the indicated* Waves *model*

***Figure 3.2:** Delete the undesired models from the project tree*

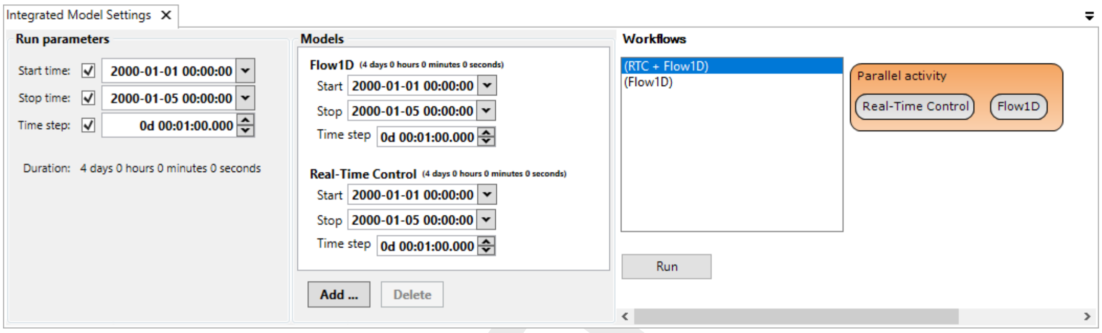
- ➢ Right click on *Rainfall Runoff* or *Waves* and select *Delete* from the context menu.
- ➢ Confirm it by pressing *OK*.
- ➢ Confirm it by pressing *Yes*.

In the central window the **Integrated Model Settings** dialog appear (Figure 3.3).

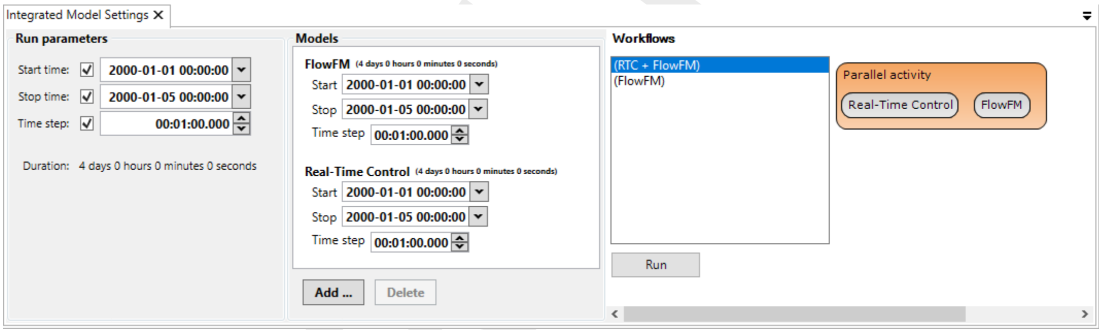
Set the start, stop and run time in **Run Parameters** tab:

- ➢ Start time “2000-01-01 00:00:00”.
- ➢ Stop time “2000-01-05 00:00:00” and
- ➢ Time step to “00:01:00.000” (one minute)

The window should now look like as in Figure 3.3.



(a) Coupling between D-Flow 1D and D-RTC



(b) Coupling between D-Flow FM and D-RTC

***Figure 3.3: Integrated model**, settings for a coupled simulation*

The D-Flow 1D + D-RTC model and the D-Flow FM + D-RTC model will be discussed in section 3.3 and section 3.4 separately.

3.3 The D-Flow 1D model

Create a D-Flow 1D model by performing the following actions, where is assumed that you are familiar with creating a one-dimensional D-Flow 1D model (SOBEK 3 User Manual, 2014).

➢ Load the model <drtc_tutorial_dflow1d.dsproj> from the tutorial directory <.../Tutorial_D-RTC/dflow1d-drtc>
➢ (Optional) Enable OpenStreetMap WMS layer. If choosen: zoom-in to the area as shown by Figure 3.4
➢ Add one branch with a length of approximately 40 km to the network. From river side to sea side. No need to adjust the custom length.
➢ Change the *Initial water depth* to “10 m” (Initial conditions in the **project** panel).
➢ Add a cross-section of type YZ with a width according Table 3.1 at chainage: ≈ 28 km

***Table 3.1:** Cross-section values of the YZ-table*

Y	Z
0	0
10	-10
490	-10
500	0

➢ Change the thalweg in the property editor to 250 m.
➢ Add an observation point at chainage: ≈ 4.5 km and call it “Brug” (i.e. at or near the ’Willemsbrug’).

- ➢ Add a second observation point at chainage: $\approx$ 21 km and call it “Tunnel” (i.e. at or near the ’Maasdeltatunnel’).
- ➢ Add a Simple Weir at chainage: $\approx$ 31 km, and call it “Kering” (i.e. at or near the ’Maeslantkering’).
- ➢ Change the *Crest width* to “500 m”.

The flow model schematisation (<network>) then should look more or less like the one given by Figure 3.4.

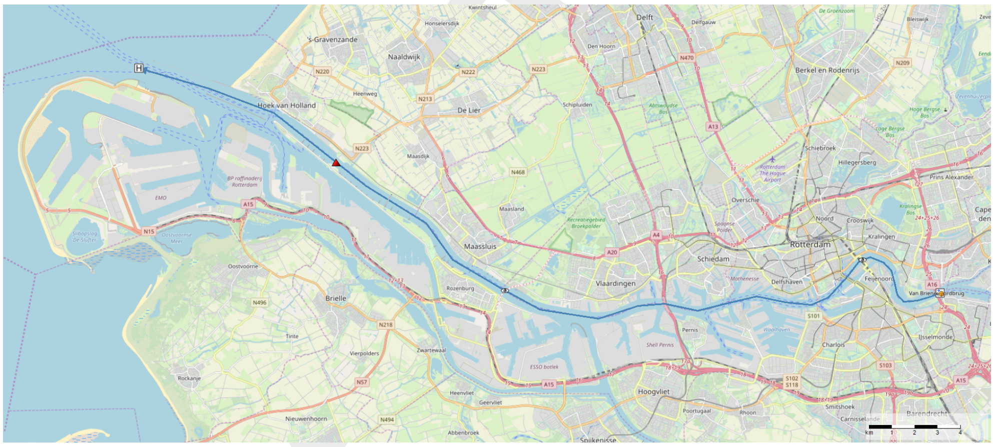


Figure 3.4: *Example water flow model schematisation with an OpenStreet background map (http:// openstreetmap.org)*

- ➢ Create a computational grid (default grid spacing).
- ➢ Set a constant water level of “–2 m” at the downstream boundary (sea side).
- ➢ Set a discharge time series boundary condition at the upstream boundary (river side) of the model (Table 3.2).

Table 3.2: *Discharge boundary condition table for the upstream end*

Date	**Discharge [m^3 s^{-1}]**
2000-01-01 00:00:00	1500
2000-01-02 00:00:00	1500
2000-01-05 00:00:00	2500

- ➢ Set the structure output parameters (in the projec tree choose ... $\rightarrow$ *Models*$\rightarrow$ *Flow1D* $\rightarrow$ *Output* $\rightarrow$ *Structures*) *Discharge*, *Crest level*, *Crest width*, *Water level up*, *Water level down*, *Head difference* and *Water level at crest* to “Current”. Other output parameters remain at default.
- ➢ Validate the model.
- ➢ Save the project.
- ➢ Run the model.

3.3.1 A simple control flow

To define a control flow a control group need to be added to the D-RTC model.

3.3.1.1 Add a Control Group

Start adding a control group by:

- Right-mouse click on <Control Groups> in the **Project** window.
- Select *Add New Control Group. . .* .

A menu (Figure 3.5) appears where the user can choose between a set of default available control groups.

- Select *Empty group*.

Control Group 1 is now added to the set of Control Groups in the **Project** window. A **Control Group** window is shown and currently empty.

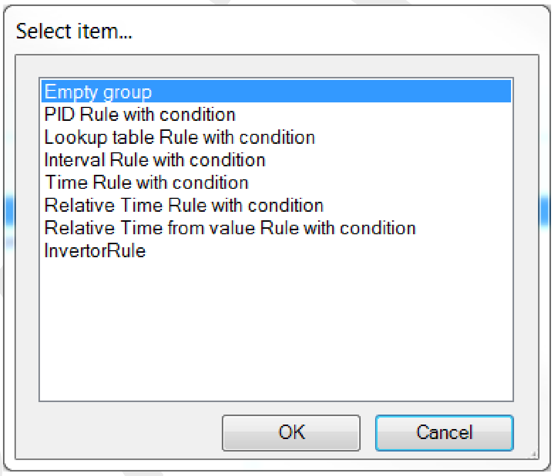


Figure 3.5: *Options for default controlgroups*

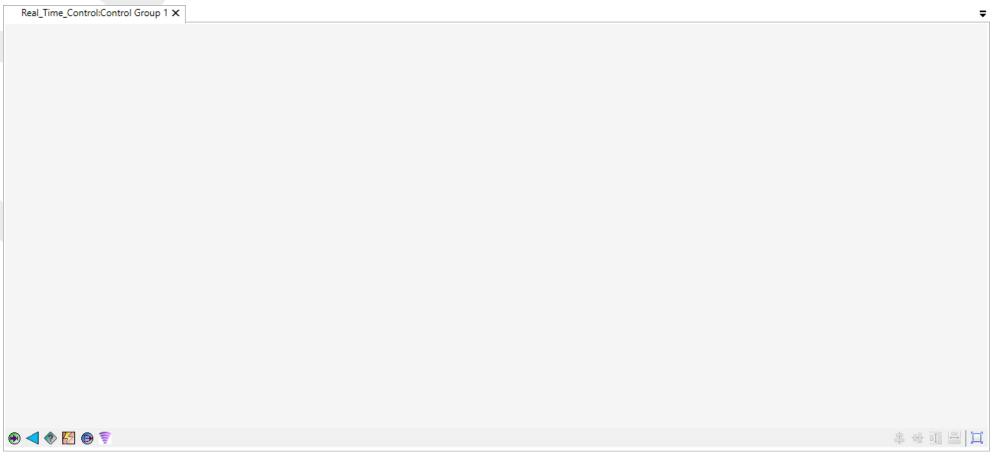


Figure 3.6: *Empty **Control Group** window*

The empty **Control Group** window contains some icons in the lower left corner of the window. In this tutorial not all icons will be used to construct a control flow, but for completeness they are listed here:

 Add an input location to the control flow (section 4.1).

 Add an mathematical expression to the control flow (section 6.2.2).

 Add a hydro (section 4.2.1) or time condition (section 4.2.2) to the control flow.

 Add a rule to the control flow (section 4.3).

 Add an output location to the control flow (section 4.4).

Add a signal to the control flow (section 4.5).

3.3.2 Construct a minimal control flow

To construct a control flow, first add an output location to the flow chart in the **Control Group** window.

- Select (in the lower left corner of the **Control Group** window).
- Click some where in the **Control Group** window.

This tool adds an output location to the flow chart presented in the **Control Group** window. Secondly add a rule to the flow chart.

- Select .
- Click some where in the **Control Group** window.

Connect the two objects to obtain a flow chart as shown in Figure 3.7.

- Move the mouse over the rule object, left-click on the anchor point of the rule object on its right side, hold the mouse clicked and find the anchor point on the left side of the output location object. Release the mouse button.

Figure 3.7: *Minimum flow chart with a PID rule*

Now set the crest level of the weir as output location.

- Right-click click on the output location object and navigate through the menus. From the available locations in the network of the flow model select the weir as location and *crest level* as controlled parameter.

The output location ellipsoid turns blue after having specified the parameter.

The default rule is a PID rule. In this tutorial we use a Time rule, because this is the simplest one.

- Right-mouse click on the rule.
- Select *Convert PID Rule to* and choose *Time Rule*.

The data of the time rule can be edited in the **Properties** window, property **Data**.

- Click on *Time Series*.
- Click than on the ellipsis button to open a table window.

Fill in the data from Table 3.3. Use the arrow keys to switch between cells and the return key to add a new line.

Table 3.3: Time Rule data for crest level

Date	Crest level [m]
2000-01-01 00:00:00	1
2000-01-02 00:00:00	1
2000-01-05 00:00:00	-8

- ➢ Set Interpolation to *linear* (default value).
- ➢ Set Extrapolation to *constant* (default value).
- ➢ Save the project.

3.3.3 Perform a simulation

To perform a coupled flow simulation

- ➢ Right-click on the <integrated model> in the **Project** window.
- ➢ Choose *Run Model*.

After running the model save the project.

- ➢ Save the project with a new name.

3.3.4 View the simulation results

Both the D-Flow 1D model and D-RTC run simultaneously and exchange data with each other during run-time on a time step basis. Figure 3.8 shows the **Project** window after the simulation run. Both D-RTC and D-Flow 1D generate their own output; the output time step of the D-Flow 1D model is set by the user (here one minute). The D-RTC model uses this time step and puts out the values of the controlled parameters per time step. Note that the output of D-RTC is the input for the next time step D-Flow 1D computes.

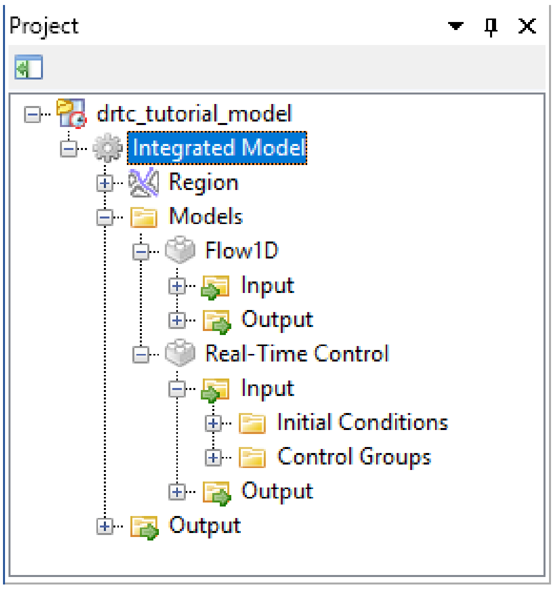


***Figure 3.8: Project** window after a coupled simulation with D-Flow 1D and D-RTC.*

3.3.4.1 View the history time series

- ➢ Select both observation point on the map,

➢ press the right mouse button and select *Query Time Series* and
➢ select the *Water level (op)*.

The window Figure 3.9 will appear.

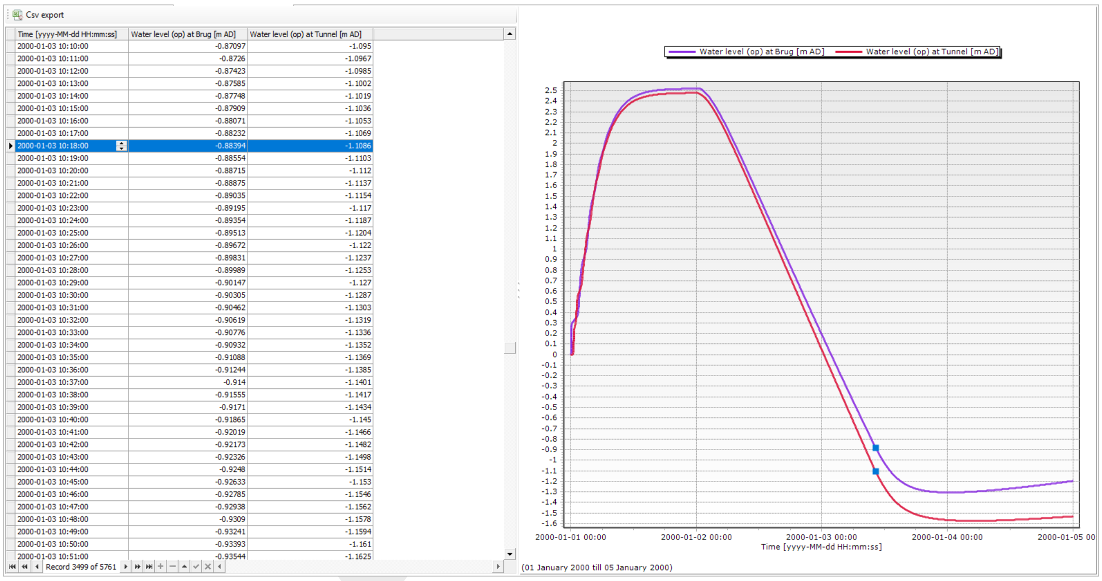


Figure 3.9: *D-Flow 1D + D-RTC time series results for the locations* Brug *and* Tunnel

3.3.4.2 Side-view

Before you can create a side-view along a branch you have to define a route along which the side-view will be shown:

➢ Select  in the ribbon.

Now you are able to create a network route along the branch.

➢ Click near the start of the branch (river side)
➢ Click near the end of the branch (sea side).

A route between the two points is created (Figure 3.10).

Figure 3.10: *Route shown on map*

➢ Click in the ribbon to open the side-view.

➢ Navigate through the time steps with the time series navigator to explore the simulation results in time along the chainage of the channel.

The created route can be deleted in **Region Contents** under <network> <Routes>.

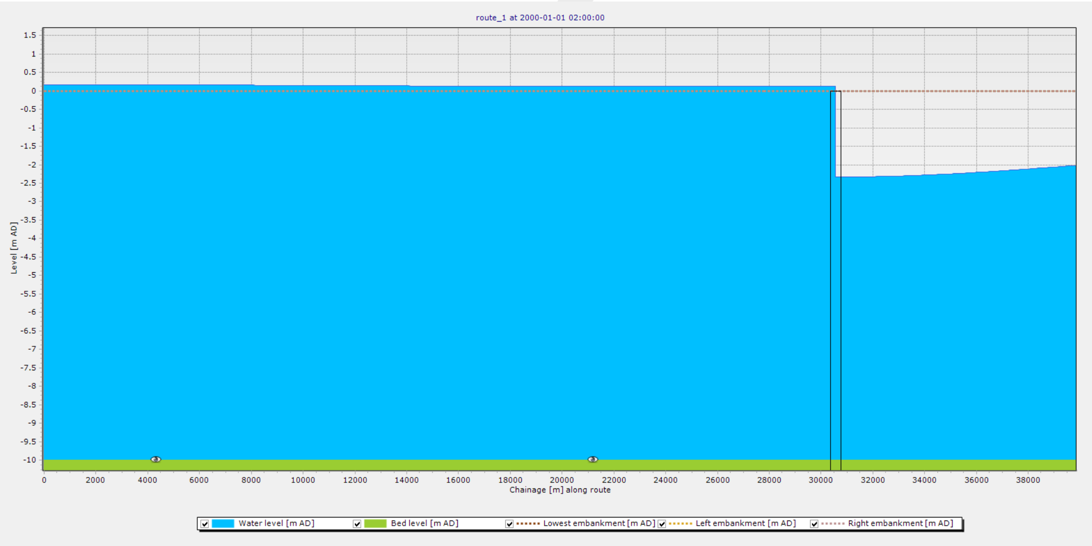


Figure 3.11: *Sideview with water level and crest level of the structure*

3.3.5 A more complex control flow

3.3.5.1 Multiple controlled parameters on one structure

Save the project under a new name, if not already done.

➢ Open the editor for *Control Group 1*.
➢ Select .
➢ Click in the flow chart to add an output location.
➢ Select the output location and right-mouse click to specify *Weir1* as location and *crest width* as controlled parameter.
➢ Select .
➢ Click in the flow chart to add a rule.
➢ Select the new rule and right-mouse click to convert the standard PID rule into a Time rule.
➢ Rename the *rule02* to *Time Rule Width*.

Fill in the values from Table 3.4 under *TimeSeries* in the **properties** window. The flow chart now looks as in Figure 3.12.

Table 3.4: *Time series for the crest width (rule 2)*

Date	Crest width [m]
2000-01-01 00:00:00	125
2000-01-02 00:00:00	125
2000-01-05 00:00:00	500

➢ Connect these two objects.

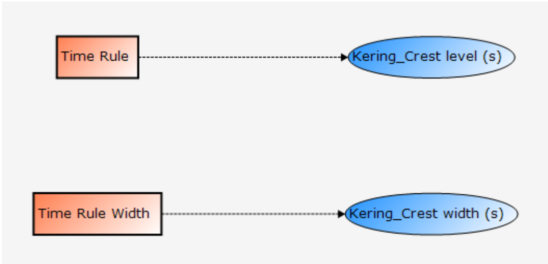


Figure 3.12: *Flowchart for example with two controlled parameters for one weir.*

Run the integrated model again.

➢ Choose *Run Model*.
➢ Save the project with a new name.

To visualize the simulation results

➢ Select the structure 'Kering'on the map.

➢ Select the *Query Time Series* tool .
➢ Press the *Control* key on your keyboard and choose "Crest level (s)" and "Water level at crest (s)" from the *Flow1D* model as output coverage (Figure 3.13).

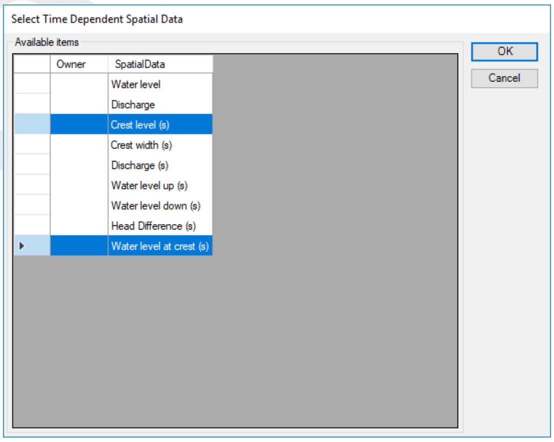
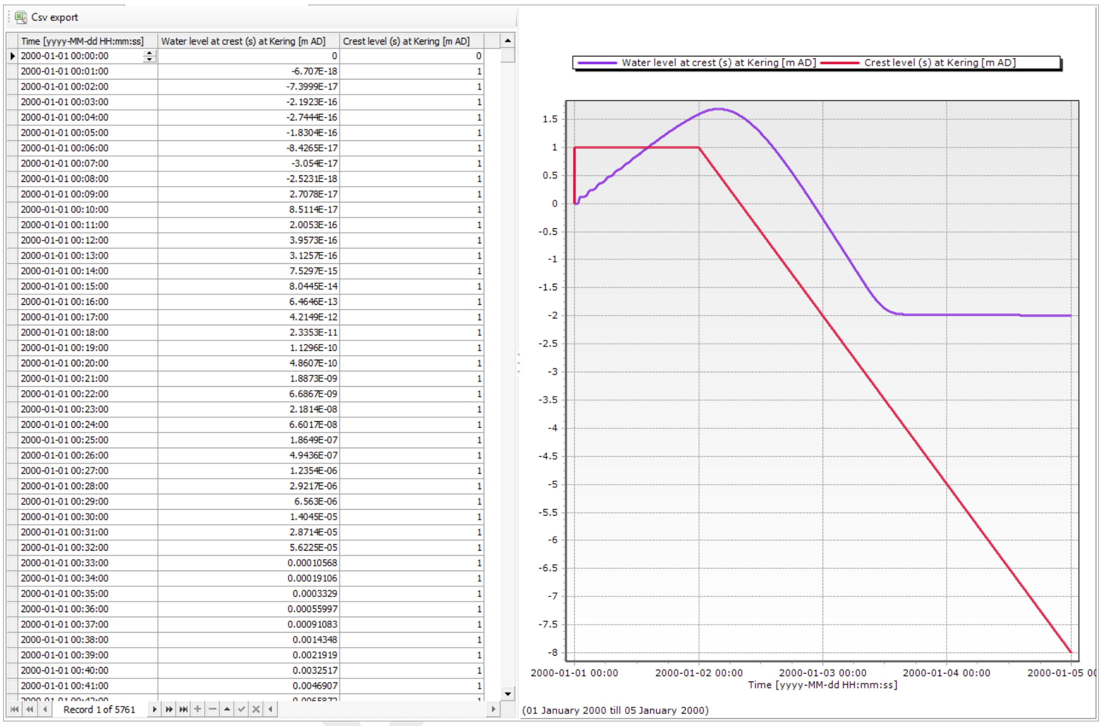


Figure 3.13: *Select output coverage*

➢ Press *OK*

The selected parameters are now plotted over time for the selected Weir Node (Figure 3.14).

➢ Click with the mouse in the diagram or select rows in the table.



Csv export

Time [yyyy-MM-dd HH:mm:ss]	Water level at crest (s) at Kering [m AD]	Crest level (s) at Kering [m AD]
2000-01-01 00:00:00	0	0
2000-01-01 00:01:00	-6.707E-18	1
2000-01-01 00:02:00	-7.3999E-17	1
2000-01-01 00:03:00	-2.1923E-16	1
2000-01-01 00:04:00	-2.7444E-16	1
2000-01-01 00:05:00	-1.8304E-16	1
2000-01-01 00:06:00	-8.4265E-17	1
2000-01-01 00:07:00	-3.054E-17	1
2000-01-01 00:08:00	-2.5231E-18	1
2000-01-01 00:09:00	2.7078E-17	1
2000-01-01 00:10:00	8.5114E-17	1
2000-01-01 00:11:00	2.0053E-16	1
2000-01-01 00:12:00	3.9573E-16	1
2000-01-01 00:13:00	3.1257E-16	1
2000-01-01 00:14:00	7.5297E-15	1
2000-01-01 00:15:00	8.0445E-14	1
2000-01-01 00:16:00	6.4646E-13	1
2000-01-01 00:17:00	4.2149E-12	1
2000-01-01 00:18:00	2.3353E-11	1
2000-01-01 00:19:00	1.1296E-10	1
2000-01-01 00:20:00	4.8607E-10	1
2000-01-01 00:21:00	1.8873E-09	1
2000-01-01 00:22:00	6.6867E-09	1
2000-01-01 00:23:00	2.1814E-08	1
2000-01-01 00:24:00	6.6017E-08	1
2000-01-01 00:25:00	1.8649E-07	1
2000-01-01 00:26:00	4.9436E-07	1
2000-01-01 00:27:00	1.2354E-06	1
2000-01-01 00:28:00	2.9217E-06	1
2000-01-01 00:29:00	6.563E-06	1
2000-01-01 00:30:00	1.4045E-05	1
2000-01-01 00:31:00	2.8714E-05	1
2000-01-01 00:32:00	5.6225E-05	1
2000-01-01 00:33:00	0.00010568	1
2000-01-01 00:34:00	0.00019106	1
2000-01-01 00:35:00	0.0003329	1
2000-01-01 00:36:00	0.00055997	1
2000-01-01 00:37:00	0.00091083	1
2000-01-01 00:38:00	0.0014348	1
2000-01-01 00:39:00	0.0021919	1
2000-01-01 00:40:00	0.0032517	1
2000-01-01 00:41:00	0.0046907	1

Figure 3.14: Crest level (s) *and* Water level at crest (s) *over time for the weir. A point in time has been selected in the diagram, the corresponding line is selected in the table.*

3.3.5.2 Multiple controlled structures

Save the project under a new name, if not already done.

➢ Add a second simple weir to the network of the water flow model at chainage ≈ 14 km, and call it 'Bolder'.

The network should now look like as given in Figure 3.15.

Figure 3.15: *D-Flow 1D network with two weirs.*

➢ Change the crest level to "–4 m".

➢ Change the crest width to "500 m".

Add a new control group to the D-RTC model

- Right-click on <Control Groups> in the **Project** window.
- Select *Empty group*.

Add an output location and a rule to the new Control Group (buttons and and convert from *PID Rule* to *Time Rule*).

Note: Within one Control Group the names have to be unique.

Fill in the table of the Time Rule properties with the data from Table 3.5.

Table 3.5: *Time series of crest level for a second structure*

Date	Crest level [m]
2000-01-01 00:00:00	-4
2000-01-02 00:00:00	-4
2000-01-05 00:00:00	-9

- Select the output location and set it to *Bolder* and *Crest level*.
- Connect the *Time Rule* object with the *Output location* object.

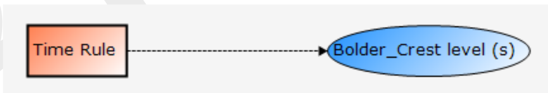
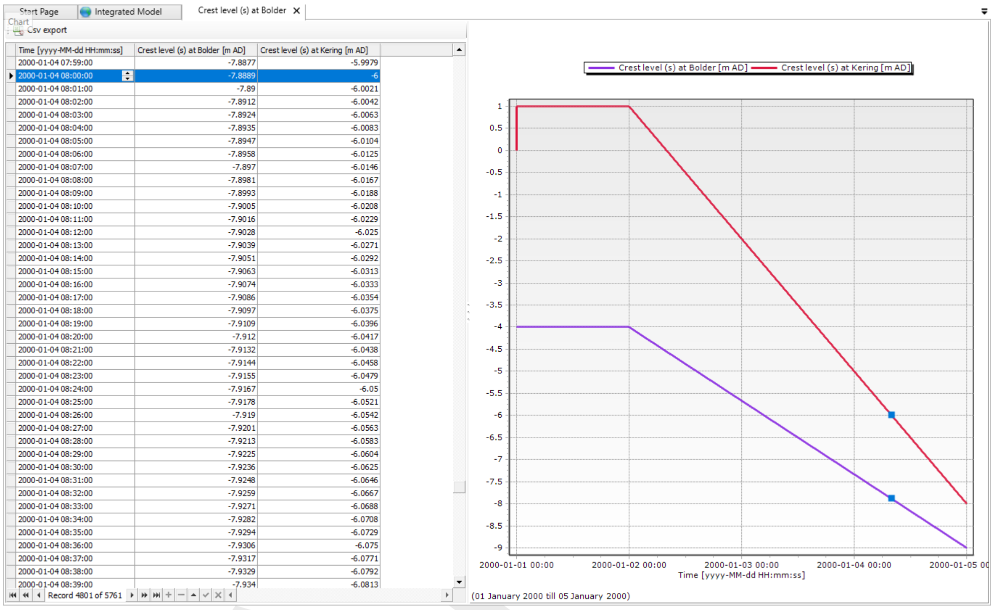


Figure 3.16: *Flow chart for* Control group 2

- Run the integrated model.
- Save the project.

- Select both wiers on the map.
- Press the right mouse button.
- Select Query time series.
- Select *Crest level (s)*.
- Press *OK*.

Apply a custom filter on the value column to find out when the crest level of 'Kering' is equal to –6 m, Figure 3.17).



Time [yyyy-MM-dd HH:mm:ss]	Crest level (s) at Bolder [m AD]	Crest level (s) at Kering [m AD]
2000-01-04 07:59:00	-7.8877	-5.9979
2000-01-04 08:00:00	-7.8889	-6
2000-01-04 08:01:00	-7.89	-6.0021
2000-01-04 08:02:00	-7.8912	-6.0042
2000-01-04 08:03:00	-7.8924	-6.0063
2000-01-04 08:04:00	-7.8935	-6.0083
2000-01-04 08:05:00	-7.8947	-6.0104
2000-01-04 08:06:00	-7.8958	-6.0125
2000-01-04 08:07:00	-7.897	-6.0146
2000-01-04 08:08:00	-7.8981	-6.0167
2000-01-04 08:09:00	-7.8993	-6.0188
2000-01-04 08:10:00	-7.9005	-6.0208
2000-01-04 08:11:00	-7.9016	-6.0229
2000-01-04 08:12:00	-7.9028	-6.025
2000-01-04 08:13:00	-7.9039	-6.0271
2000-01-04 08:14:00	-7.9051	-6.0292
2000-01-04 08:15:00	-7.9063	-6.0313
2000-01-04 08:16:00	-7.9074	-6.0333
2000-01-04 08:17:00	-7.9086	-6.0354
2000-01-04 08:18:00	-7.9097	-6.0375
2000-01-04 08:19:00	-7.9109	-6.0396
2000-01-04 08:20:00	-7.912	-6.0417
2000-01-04 08:21:00	-7.9132	-6.0438
2000-01-04 08:22:00	-7.9144	-6.0458
2000-01-04 08:23:00	-7.9155	-6.0479
2000-01-04 08:24:00	-7.9167	-6.05
2000-01-04 08:25:00	-7.9178	-6.0521
2000-01-04 08:26:00	-7.919	-6.0542
2000-01-04 08:27:00	-7.9201	-6.0563
2000-01-04 08:28:00	-7.9213	-6.0583
2000-01-04 08:29:00	-7.9225	-6.0604
2000-01-04 08:30:00	-7.9236	-6.0625
2000-01-04 08:31:00	-7.9248	-6.0646
2000-01-04 08:32:00	-7.9259	-6.0667
2000-01-04 08:33:00	-7.9271	-6.0688
2000-01-04 08:34:00	-7.9282	-6.0708
2000-01-04 08:35:00	-7.9294	-6.0729
2000-01-04 08:36:00	-7.9306	-6.075
2000-01-04 08:37:00	-7.9317	-6.0771
2000-01-04 08:38:00	-7.9329	-6.0792
2000-01-04 08:39:00	-7.934	-6.0813

Figure 3.17: *Crest levels for 'Kering' and 'Bolder', selected time stamp is based on the selected crest level of –6 m for 'Kering'*

.

3.3.6 Control flows with conditions

3.3.6.1 A control flow with one condition

First we will cleanup the existing project.

➢ Save the project under a different name, if not already done.
➢ Remove the second weir.

The network should now look like Figure 3.18.

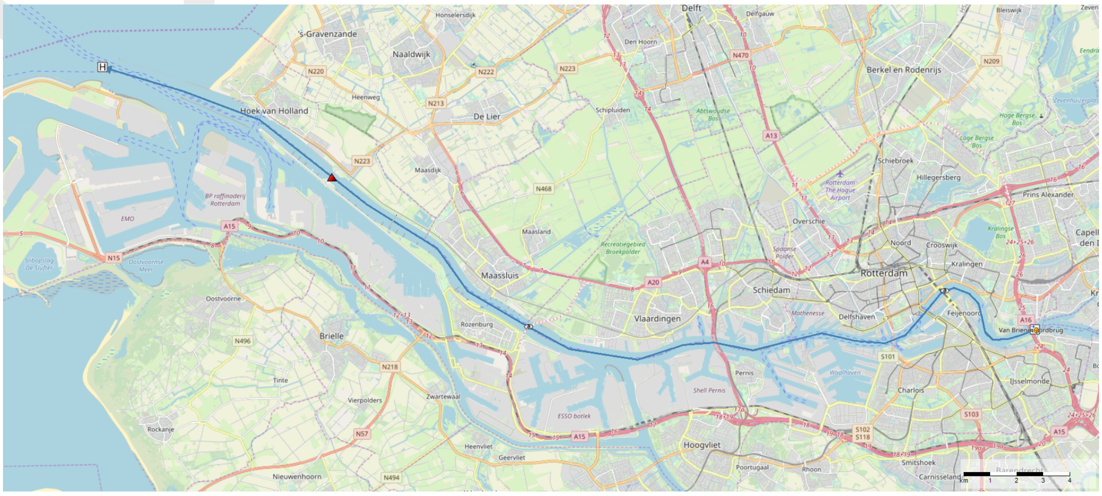


Figure 3.18: *D-Flow 1D network with one weir and a two observation points.*

➢ Remove Control Group 2 from the D-RTC model.
➢ Save the project.

Go to Control Group 1. Let the Rule control the crest level of the weir.

- ➢ Add a condition by selecting and a mouse-click in the flowchart.
- ➢ Connect the right-side of the condition (true-output) to the left side of the time rule.

Note: The condition is now shown with a thick, black line instead of the rule, indicating that the control flow now starts with the condition instead of the rule.

- ➢ Select the condition and fill in the **Properties** window with data from Table 3.6. This is a so-called Hydro Condition, which evaluates input data and puts out true or false.

Table 3.6: *Parameter-Data table for condition*

Operation	>
Value	3

Add an input location to the Control Flow.

- ➢ Select and mouse-click in the flowchart.
- ➢ Select observation point 'Brug' as input location and the *Water level (op)* as control parameter.
- ➢ Connect the bottom anchor point of the data location object to the top anchor point of the condition.

The flowchart now looks like Figure 3.19.

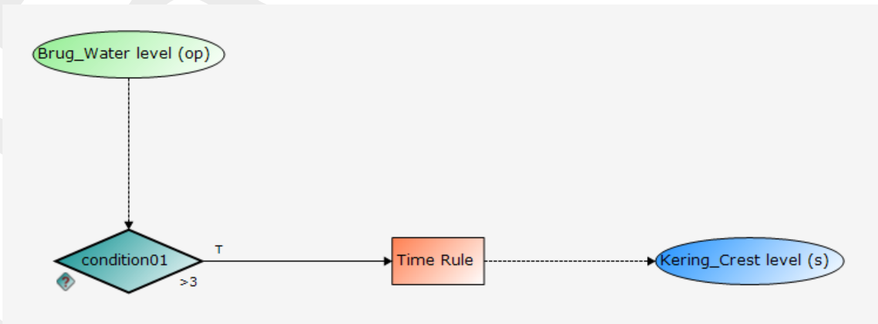


Figure 3.19: *Flowchart with a Hydro Condition and a Time Rule. The rule is connected with the true-output of the condition.*

The condition now checks whether the water level in the observation point is higher than 3 m. If this is the case, the condition returns 'true', which activates the rule. If this is not the case, the condition returns 'false' as result, which means that the rule is not activated.

- ➢ Run the integrated model.
- ➢ Save the project.

3.3.6.2 A control flow with two conditions: logical AND

Save the project under a new name, if not already done.

Add a new hydro condition to the flow chart and fill in the **Properties** window with the data given in Table 3.7. Add a input location object, select observation point 'Brug' and choose *Discharge (op)* as control parameter. Connect the true-output of the new condition with the input of the first condition. The flowchart now looks like Figure 3.20.

Table 3.7: Parameter-Data table for the second condition

Operation	>
Value	300

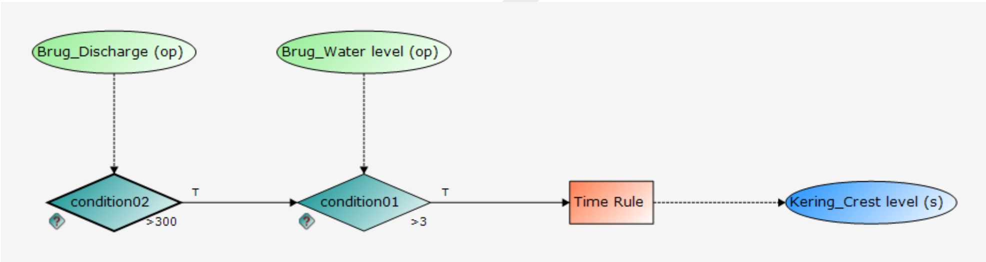


***Figure 3.20:** Flowchart with two Hydro conditions in an AND combination combined with a Time rule.*

This flow chart represents a logical AND combination of two conditions. The rule is only active if both the first and the second condition are active.

- ➢ Run the integrated model.
- ➢ Save the project.

Run the integrated model and analyze the simulation results. Check when the rule is active and the status of the conditions during simulation time.

3.3.6.3 A control flow with two conditions: logical OR

Save the project under a new name.

Select the connection between the two conditions and delete it with the Delete-key on your keyboard. Connect the bottom anchor point of *condition01* (the false-output of the condition that evaluates the water level) to the left side of *condition02* (evaluates discharge). Then connect the right side anchor point of *condition02* to the left anchor point of the rule. Arrange the elements in such a way that the flowchart looks like Figure 3.21.

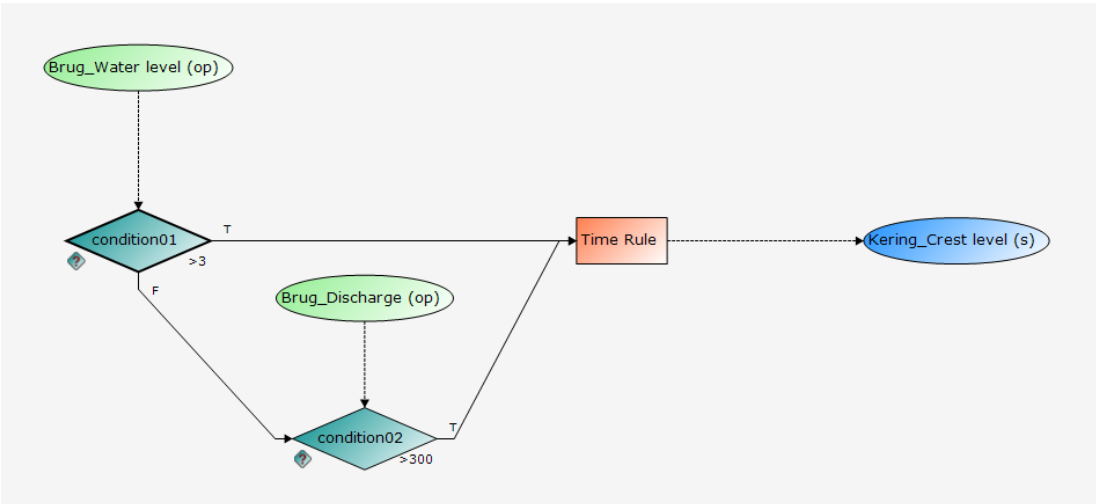


***Figure 3.21:** Flowchart with two Hydro conditions in an OR combination and a Time rule.*

This is an example of two conditions in an OR combination. Whereas in Figure 3.20 both conditions had to be true for the rule to be active, in Figure 3.21 only one of the two conditions needs to be true for the rule to be active.

Run the integrated model, save the project and analyze the results.

➢ Run the integrated model.
➢ Save the project.

3.4 The D-Flow FM model

Create a simple D-Flow FM model by performing the following actions, where is assumed that you are familiar with creating a two-dimensional D-Flow FM model (Deltares, 2024).

➢ Load the model <drtc_tutorial_dflowfm.dsproj> from the tutorial directory <. . . /Tutorial_D-RTC/dflowfm-drtc>
➢ (Optional) Enable OpenStreetMap WMS layer. If choosen: zoom-in to the area as shown by Figure 3.22
➢ Add an observation point ≈ 4500 m from the river boundary and call it “Brug” (i.e. at or near the ’Willemsbrug’).
➢ Add an observation point ≈ 21 500 m from the river boundary and call it “Tunnel” (i.e. at or near the ’Maasdeltatunnel’).
➢ Add a *Structure* with type *Simple Gate* ≈ 8.5 km from the sea boundary and call it “Kering” (i.e. at or near the ’Maeslantkering’).
➢ Change the *Crest width* to “500 m”.
➢ Change the *Gate lower edge level* to “100 m”.

The flow model schematisation then should look more or less like the one given by Figure 3.22.

Figure 3.22: *Example water flow model schematisation with an OpenStreet background map (http:// openstreetmap.org)*

➢ Set a constant water level of “–2 m” at the sea boundary.
➢ Set a discharge time series boundary condition at the river boundary of the model (Table 3.8).

Table 3.8: *Discharge boundary condition table for the river boundary*

Date	Discharge [m^3 s^{-1}]
2000-01-01 00:00:00	1500
2000-01-02 00:00:00	1500
2000-01-05 00:00:00	2500

- ➢ Save the project.
- ➢ Validate the model.
- ➢ Run the model.

3.4.1 A simple control flow

To define a control flow a control group need to be added to the D-RTC model.

3.4.1.1 Add a Control Group

Start adding a control group by:

- ➢ Right-mouse click on <Control Groups> in the **Project** window.
- ➢ Select *Add New Control Group. . .* .

A menu (Figure 3.23) appears where the user can choose between a set of default available control groups.

- ➢ Select *Empty group*.

Control Group 1 is now added to the set of Control Groups in the **Project** window. A **Control Group** window is shown and currently empty.

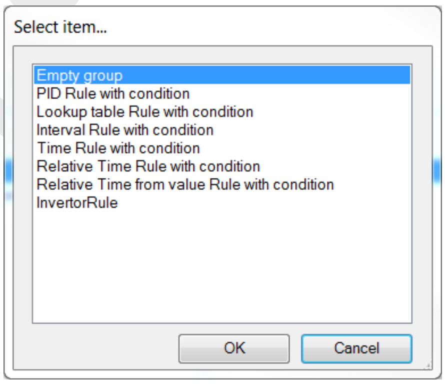


Figure 3.23: *Options for default control groups*

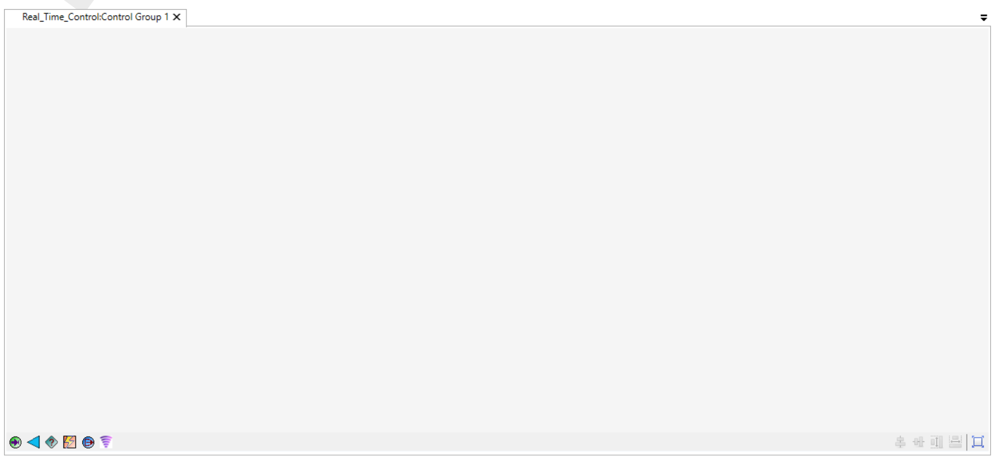


Figure 3.24: *Empty **Control Group** window*

The empty **Control Group** windows contains some icons in the lower left corner of the window.

In this tutorial not all icons will be used to construct a control flow, but for completeness they are listed here:

	Add an input location to the control flow (section 4.1).
	Add a mathematical expression to the control flow (section 6.2.2).
	Add a hydro (section 4.2.1) or time condition (section 4.2.2) to the control flow.
	Add a rule to the control flow (section 4.3).
	Add an output location to the control flow (section 4.4).
	Add a signal to the control flow (section 4.5).

3.4.1.2 Construct a minimal control flow

To construct a control flow, first add an output location to the flow chart in the **Control Group** window.

➢ Select  (in the lower left corner of the **Control Group** window).
➢ Click some where in the **Control Group** window.

This tool adds an output location to the flow chart presented in the **Control Group** window. Secondly add a rule to the flow chart.

➢ Select .
➢ Click some where in the **Control Group** window.

Connect the two objects to obtain a flow chart as shown in Figure 3.25.

➢ Move the mouse over the rule object, left-click on the anchor point of the rule object on its right side, hold the mouse clicked and find the anchor point on the left side of the output location object. Release the mouse button.

Figure 3.25: *Minimum flow chart with a PID rule*

Now set the crest level of the weir as output location.

➢ Right-click click on the output location object and navigate through the menus. From the available locations in the network of the flow model select the *Structure* “Kering” as location and *CrestLevel* as controlled parameter.

The output location ellipsoid turns blue after having specified the parameter.

The default rule is a PID rule. In this tutorial we use a Time rule, because this is the simplest one.

➢ Right-mouse click on the rule and
➢ select *Convert PID Rule to* and choose *Time Rule*.

The data of the time rule can be edited in the **Properties** window, property **Data** .

➢ Click on <Time Series>.
➢ Click on the ellipsis button to open a table window.

Fill in the data from Table 3.9. Use the arrow keys to switch between cells and the return key to add a new line.

Table 3.9: *Time Rule data for crest level*

Date	Crest level [m]
2000-01-01 00:00:00	1
2000-01-02 00:00:00	1
2000-01-05 00:00:00	-8

- ➢ Close the <Time Series> table window.
- ➢ In the *Properties* window: Set Extrapolation to *constant* (default value).
- ➢ Set Interpolation to *linear* (default value).
- ➢ Save the project.

3.4.2 Perform a simulation

To perform a coupled flow simulation

- ➢ Right-click on the <integrated model> in the **Project** window.
- ➢ Choose *Run Model*.

After running the model save the project.

- ➢ Save the project with a new name.

3.4.3 View the simulation results

Both the D-Flow FM model and D-RTC run simultaneously and exchange data with each other during run-time on a time step basis. Figure 3.26 shows the **Project** window after the simulation run.

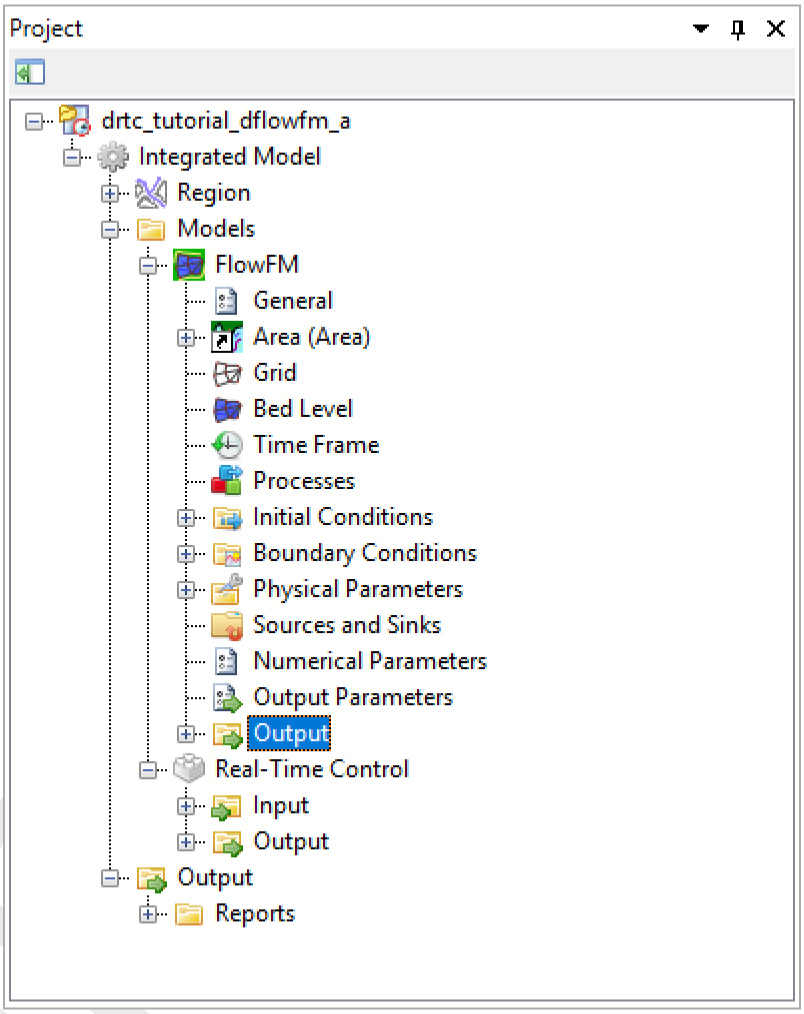


***Figure 3.26: Project** window after a coupled simulation with D-RTC and D-Flow 1D.*

Both D-RTC and D-Flow FM generate their own output. The output time step of the D-Flow FM model is set by the user. The D-RTC model uses this time step and puts out the values of the controlled parameters per time step. Note that the output of D-RTC is the input for the next time step D-Flow FM.

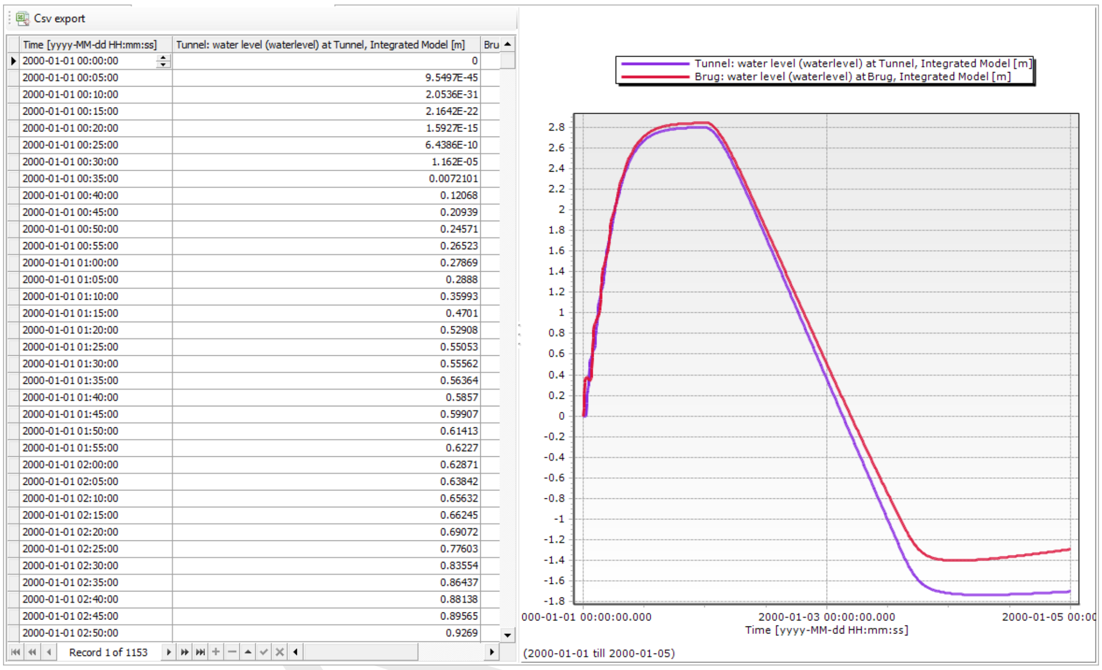


Figure 3.27: *D-Flow FM + D-RTC time series results for the locations* Brug *and* Tunnel

3.4.4 A more complex control flow

3.4.4.1 Multiple controlled parameters on one structure

Save the project under a new name, if not already done.

➢ Open the editor for *Control Group 1*.
➢ Select .
➢ Click in the flow chart to add an output location.
➢ Right click on the output location to specify *Output locations* → *Structures* → *Kering* → *GateOpeningWidth* as controlled parameter.
➢ Select .
➢ Click in the flow chart to add a rule.
➢ Select the new rule and right-mouse click to convert the standard *PID rule* into a *Time rule*.
➢ Rename the *rule01* to *Time Rule Width*.

Fill in the values from Table 3.10 under *TimeSeries* in the **properties** window. The flow chart now looks as in Figure 3.12.

Table 3.10: *Time series for the crest width (rule 2)*

Date	Crest width [m]
2000-01-01 00:00:00	125
2000-01-02 00:00:00	125
2000-01-05 00:00:00	500

➢ Connect these two objects, see Figure 3.28.

Figure 3.28: *Flowchart with a time rule for the gate opening width.*

Run the integrated model again.

➢ Choose *Run Model*.
➢ Save the project with a new name.

To visualize the simulation results

➢ Select the structure 'Kering' on the map.
➢ Select the *Query Time Series* tool from the ribbon, .
➢ Choose "gate water level up" (i.e. river side of the gate), see Figure 3.29.

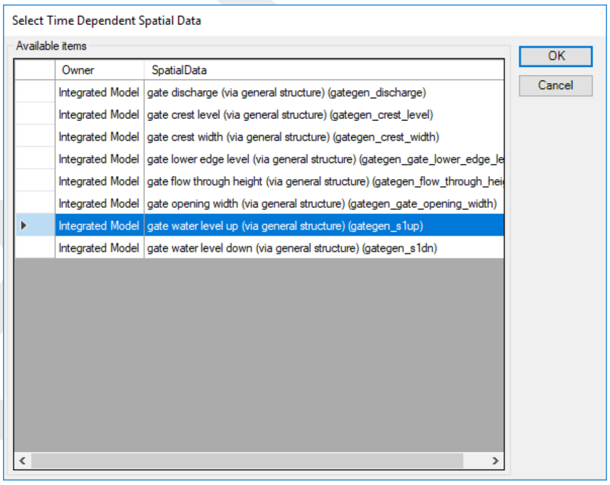


Figure 3.29: *Select time series parameters*

➢ Press *OK*

The selected parameter is now plotted over time for the selected *Gated Weir*.

➢ Click with the mouse in the diagram or select rows in the table. see Figure 3.30.

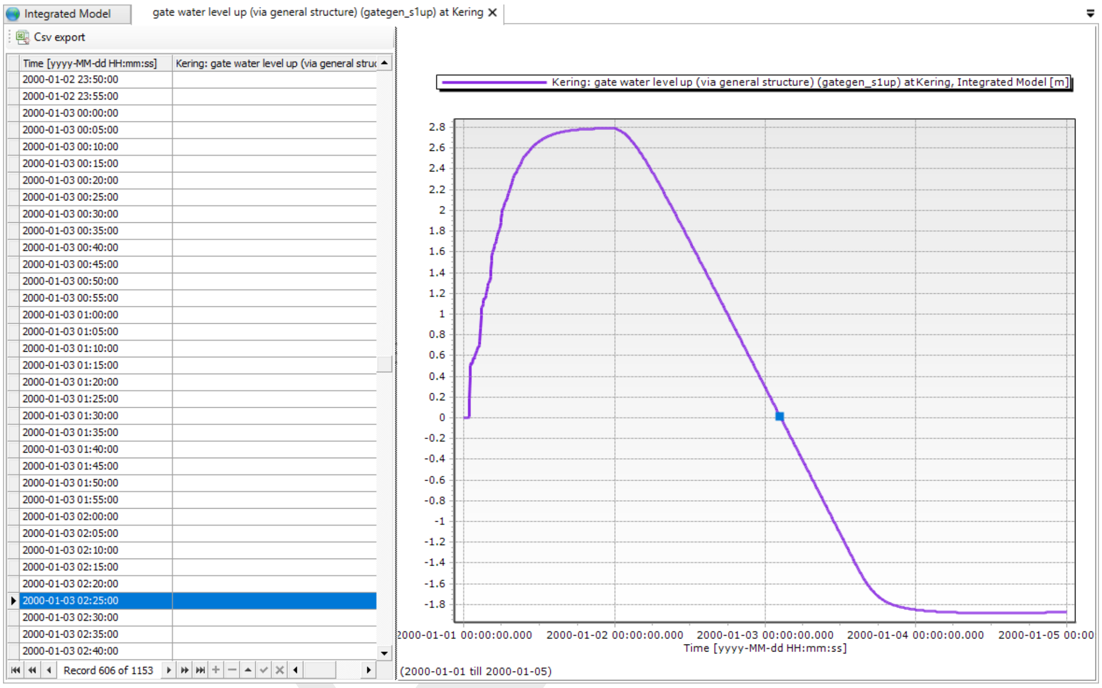


Figure 3.30: *History plot of 'Kering' of the* gate water level up *parameter. A point in time has been selected in the diagram, the corresponding line is selected in the table.*

3.4.5 Multiple controlled structures

Save the project under a new name, if not already done.

➢ Add a second *Structure* - this time type *weir* - to the model of the water flow model at ≈ 14 km from the river boundary and call it 'Bolder'.

The model should now look like the one given in Figure 3.31.

Figure 3.31: *D-Flow FM model with two structures: 'Kering' and 'Bolder'.*

➢ Change the crest level to "–4 m".
➢ Change the crest width to "500 m".
➢ Change the gate lower edge level to "100 m".

Add a new control group to the D-RTC model

➢ Right-click on <Control Groups> in the **Project** window.
➢ Select *Empty group*.

Add an output location and a rule to the new Control Group (buttons and), convert rule from *PID Rule* to *Time Rule* and rename the rule from *rule01* to *Time Rule*.

Note: Within one Control Group the names have to be unique.

Fill in the table of the Time Rule properties with the data from Table 3.11.

Table 3.11: *Time series of crest level for a second structure*

Date	Crest level [m]
2000-01-01 00:00:00	-4
2000-01-02 00:00:00	-4
2000-01-05 00:00:00	-9

- ➢ Select the output location and set it to *Weir2D* → *Bolder* → *Crest level*.
- ➢ Connect the *Time Rule* object with the *Output location* object.

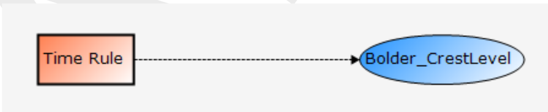


Figure 3.32: *Flow chart for* Control group 2

- ➢ Run the integrated model.
- ➢ Save the project.

- ➢ Select both *Structures* on the map.
- ➢ Press the right mouse button.
- ➢ Select Query time series.
- ➢ Select *Crest level*.
- ➢ Press *OK*.

Apply a custom filter on the value column to find out when the crest level of Weir1 is equal to –6 m, Figure 3.33).

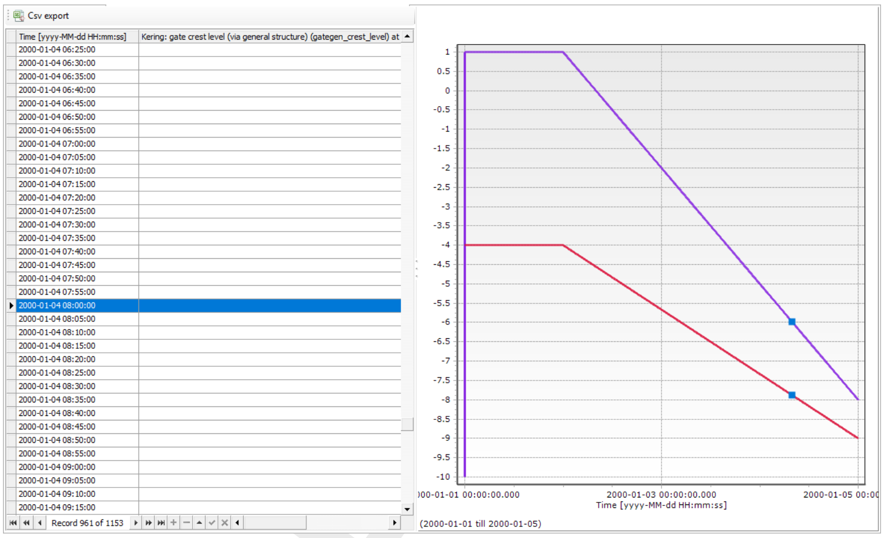


Figure 3.33: *Crest levels for gated wiers 'Kering' and 'Bolder', selected time stamp is based on the selected crest level of −6 m for 'Kering'*

.

3.4.6 A control flow with one condition

First we will cleanup the existing project.

- ➢ Save the project under a different name, if not already done.
- ➢ Remove the second *Structure* named 'Bolder'.

The model should now look like Figure 3.34.

Figure 3.34: *Example water flow model schematisation with an OpenStreet background map (http://openstreetmap.org), this model is the same as in Figure 3.22.*

- ➢ Remove Control Group 2 from the D-RTC model.
- ➢ Save the project.

Go to Control Group 1. Let the Time Rule control the crest level of the gated weir 'Kering'.

- ➢ Add a condition by selecting and a mouse-click in the flow chart.
- ➢ Connect the right-side of the condition (true-output) to the left side of the time rule.

Note: The condition is now shown with a thick, black line instead of the rule, indicating that the control flow now starts with the condition instead of the rule.

- ➢ Select the condition and fill in the **Properties** window with data from Table 3.12. This is a so-called Hydro Condition, which evaluates input data and puts out true or false.

Table 3.12: *Parameter-Data table for condition*

Operation	$>$
Value	3

Add an input location to the Control Flow.

- Select  and mouse-click in the flowchart.
- Select observation point as data location and the *Water level* as control parameter (*Input location* $\rightarrow$ *GroupableFeature2DPoint* $\rightarrow$ *Brug* $\rightarrow$ *water_level*).
- Connect the bottom anchor point of the data location object to the top anchor point of the condition.

The flowchart now looks like Figure 3.35.

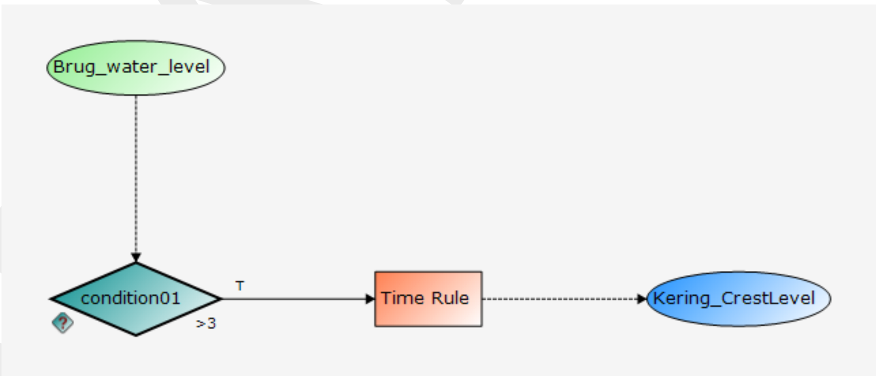


Figure 3.35: *Flowchart with a Hydro Condition and a Time Rule. The rule is connected with the true-output of the condition.*

The condition now checks whether the water level in the observation point is higher than 3 m. If this is the case, the condition returns 'true', which activates the rule. If this is not the case, the condition returns 'false' as result, which means that the rule is not activated.

- Run the integrated model.
- Save the project.
- As exercise: Analyze the results

4 All about the modelling process

The modelling process of D-RTC consist of several steps, these steps could contain the steps listed below but should have at least the step *Rule* and *Output location*:

- Add an input location to the controlflow (section 4.1).
- Add a hydro (section 4.2.1) or time condition (section 4.2.2) to the controlflow.
- Add a rule to the controlflow (section 4.3).
- Add an output location to the controlflow (section 4.4).
- Add a signal to the controlflow (section 4.5).

4.1 Input locations

For some rules the hydraulic input location need to be specified. Selecting the input location is supported by the GUI. First select the input icon ( ) and then click some where in the flow chart. In the flowchart an object will appear with the text “[Not Set]” in it. Press the right mouse button and select the input location. This can be done via several drop down list or via a table option, see Figure 4.1 or Figure 4.2.

After selecting the input location the selected item is given in the object on the flowchart, see Figure 4.3.

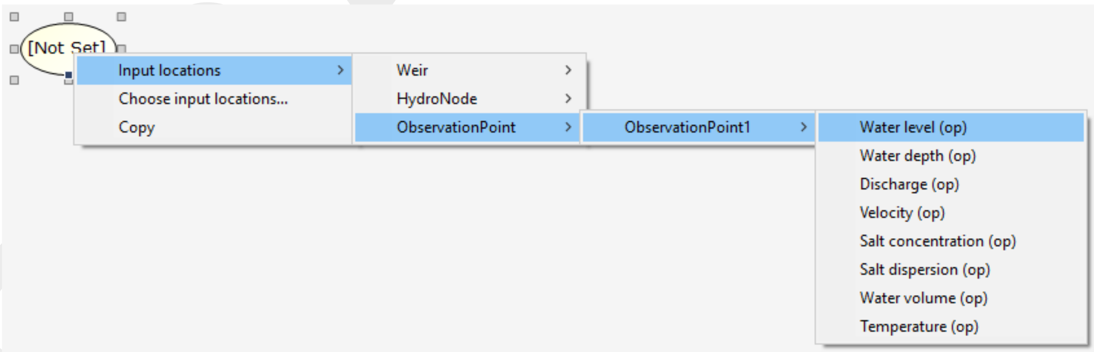


Figure 4.1: *Select input location via drop down lists*

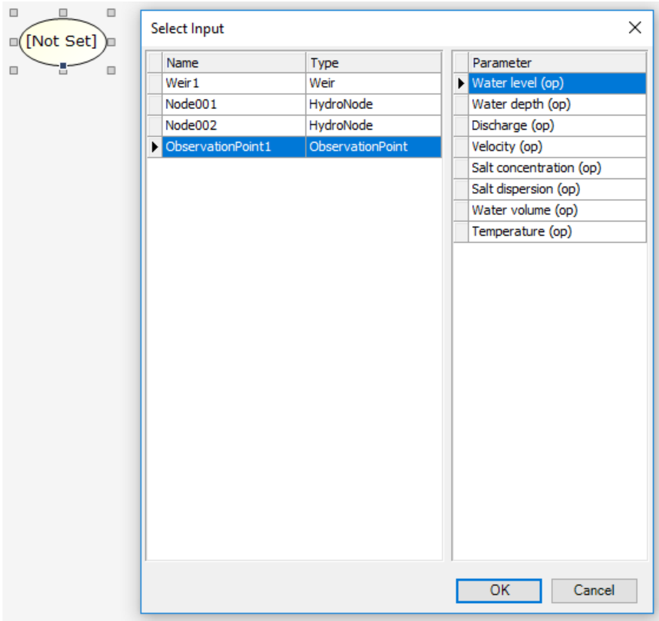


Figure 4.2: *Select input location via a table*

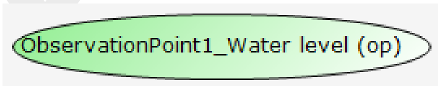


Figure 4.3: *Selected input location, as given in the object on the flowchart.*

4.2 Conditions

In D-RTC there are three types of conditions:

1. Hydro condition (section 4.2.1)
2. Time condition (section 4.2.2)
3. Standard differential condition (section 4.2.3)

The hydro condition uses data to assess whether a rule should be active or not, while the time rule uses a timetable and is therefore independent of data.

You can change the condition by pressing the right mouse button on the condition icon and select the desired condition, see Figure 4.4.

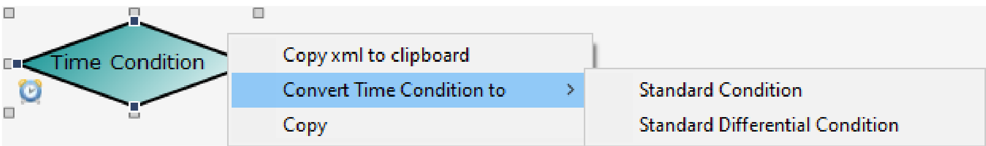


Figure 4.4: *Convert the type of condition.*

4.2.1 Hydro condition

Figure 4.5 shows an example of the use of a hydro condition in a flowchart. A hydro condition always needs input data, which are connected to the condition at the top side. These data are the control parameters and can be any parameter available at a control location, i.e. water level, discharge, velocity, but also salt concentration or water temperature, depending on the modules used.

Figure 4.6 shows the **Properties** window for a hydro condition. A hydro condition has as parameters:

- input
- Operation
- Value
- Id
- Name

The input is equal to the selected control location and parameter (in this case waterlevel at observation point 1). The value is the setpoint of the control parameter. The hydro condition checks with the operation how the actual value of the control parameter relates to the setpoint. In the example in Figure 4.6 the hydro condition checks if the waterlevel at the observation point is larger than zero. If this is the case, the operation is true and the hydro condition is also true. If this is not the case, the operation is false as is the hydro condition. The following operations are available

- $>$ (larger)
- $<$ (smaller)
- $=$ (equal)
- $<>$ (not equal)
- $>=$ (larger or equal)
- $<=$ (smaller or equal)

If the hydro condition is true, the rule connected to the true side (right side of the condition) is activated and the rule connected to the false side (bottom side of the condition) is deactivated. Otherwise, if the condition is false, the rule connected to the true side is deactivated and the rule connected to the false side is activated.

The id is obligatory and unique for the control group in which it is used. The name is not obligatory and not necessarily unique.

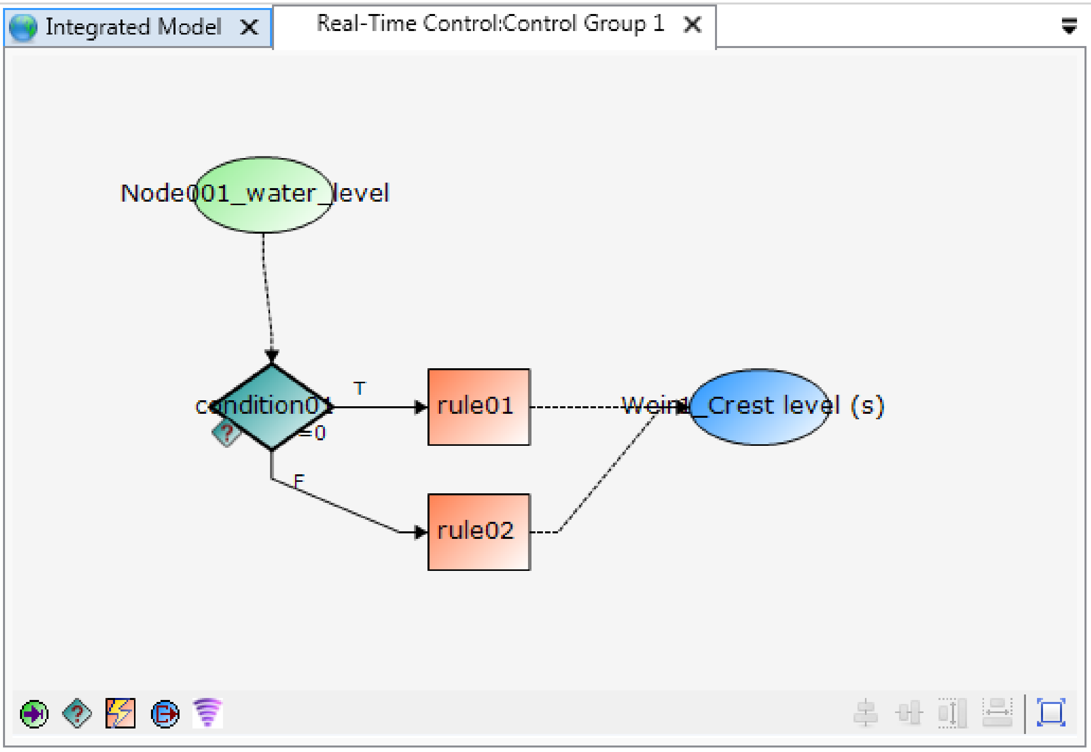


Figure 4.5: Example of a flowchart with a hydro condition

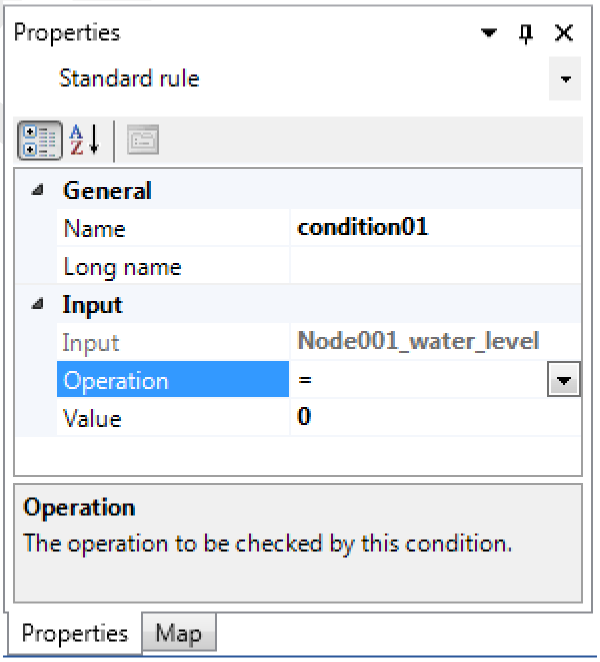


***Figure 4.6:** A hydro condition in the **Properties** Window*

4.2.2 Time condition

Figure 4.7 shows an example of the use of a time condition. The main difference in flowchart compared to a hydro condition is the absence of input data. The time condition is independent of simulation results or measurements. It only needs a time table in which is stated during which time the condition is true or false.

Figure 4.8 shows the **Properties** window for a time condition. A time condition has the following parameters:

- ◇ Time Series
- ◇ Extrapolation (constant, periodic)
- ◇ Id: obligatory, unique for each Control Group
- ◇ Name: optional and not necessarily unique

By clicking on *Time Series* and selecting ... a window pops up in which the time table can be entered. For each time entry true or false can be checked. Between two consecutive entries the value of the first time entry is maintained. If the condition is true, the rule connected to the true (right) side of the condition is activated, otherwise deactivated. Similarly, if the condition is false, the rule connected to the false (bottom) side of the condition is activated, otherwise deactivated.

The parameter extrapolation controls what happens outside the ranges of the defined timetable. Three options are possible:

- ◇ Constant: the first and last value of the timetable are used for each time before and after the defined timetable respectively.
- ◇ Periodic: the time table is repeated before its first and after its last time entry.
- ◇ None: no extrapolation, before the first and after the last time entry no rules are triggered and the value of the controlled parameter is unaffected by RTC.

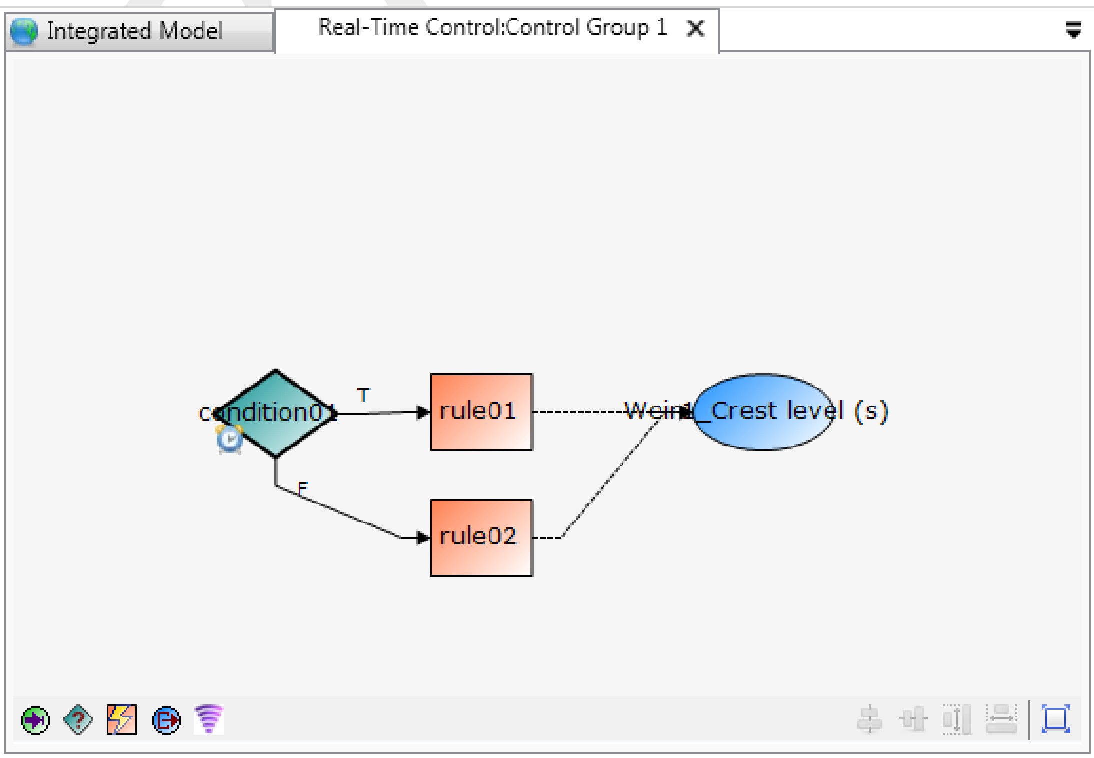


Figure 4.7: *Example of a flowchart with a time condition*

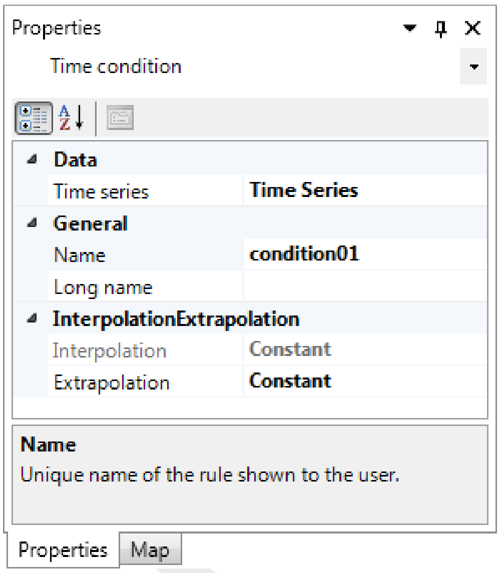


***Figure 4.8:** A time condition in the **Properties** window*

4.2.3 Standard Differential Condition

The standard differential condition checks if the data at the input location is changed (decrease, increase or not) w.r.t. the previous value.

Figure 4.9 shows an example of the use of a time standard differential condition in a flowchart. A standard differential condition always needs input data, which is connected to the condition at the top side. These data are the control parameters and can be any parameter available at a control location, i.e. water level, discharge, velocity, but also salt concentration or water temperature, depending on the modules used.

Figure 4.10 shows the **Properties** window for a differential condition condition. A differential standard condition has as input parameters:

◇ Input
◇ Operation
◇ Name
◇ Long name

The input is equal to the selected control location and parameter (in this case waterlevel at observation point 1). The standard differential condition checks with the operation how the actual value of the control parameter relates to the previous value of the parameter (not necessarily the previous time step of the hydrodynamic modue). In the example in Figure 4.10 the standard differential condition checks if the waterlevel at the observation point is increased w.r.t. the previous value. If this is the case, the operation is true and the standard differential condition is also true. If this is not the case, the operation is false as is the standard differential condition. The following operations are available

◇ Increasing
◇ Increasing or unchanged
◇ Unchanged
◇ Changed
◇ Decreasing or unchanged
◇ Decreasing

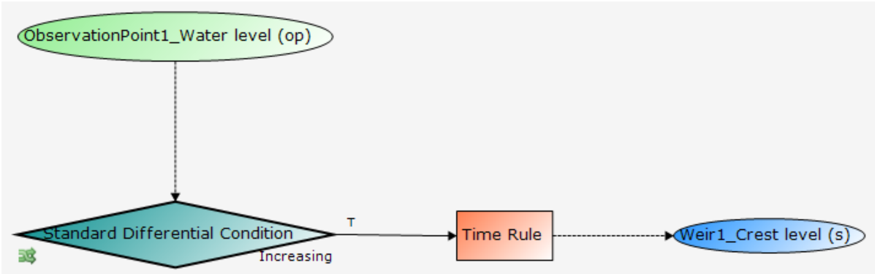


Figure 4.9: *Example of a flowchart with a standard time differential condition*

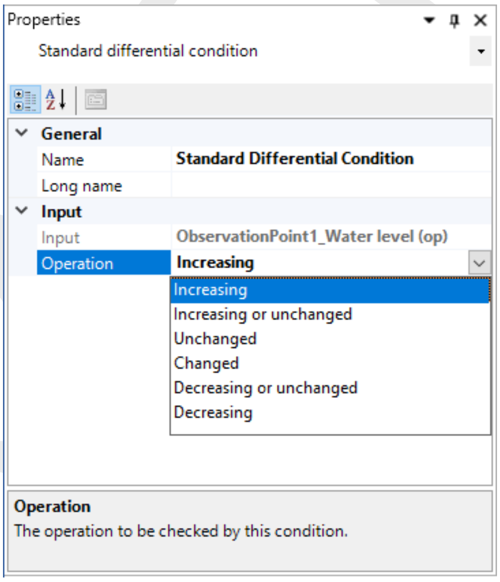


Figure 4.10: *A standard time differential condition in the* ***Properties*** *window, showing the different options*

4.3 Rules

In D-RTC there are six different types of rules:

1. Section 4.3.1, Lookup table rule
2. Section 4.3.2, Time rule
3. Section 4.3.3, PID rule
4. Section 4.3.4, Interval rule
5. Section 4.3.5, Relative from time/value rule
6. Section 4.3.6, Invertor rule

All rules will be discussed below.

4.3.1 Lookup table rule

The Lookup table rule can be used to operate a structure as a function of a control parameter, such as waterlevel or discharge at an observation point. The rule looks up the controlled parameter from a table, where a relation between the control parameter (input) and the controlled parameter (output) must be specified. An example of such a table is Table 4.1. Here the crest level of a weir (controlled parameter, output) is determined in dependence of the water level at an observation point in the model. The control parameter must be available as model output.

Table 4.1: *Example Lookup table rule for structure*

Control parameter: water level at observation point [m AD]	**Controlled parameter:** crest level of weir [m AD]
3.0	1.8929
4.0	2.9965
5.0	4.1001
6.0	5.2037

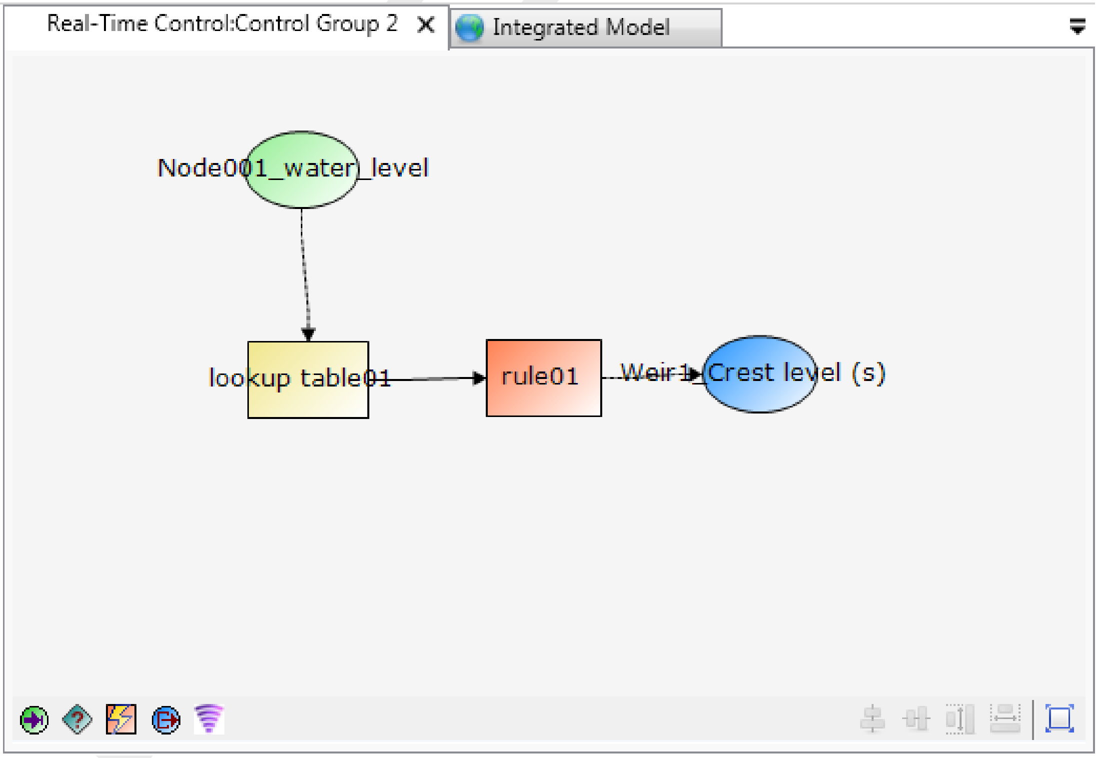


Figure 4.11: *A Lookup table rule in the flowchart (right) and on the map (left)*

Figure 4.12 shows the **Properties** window for the Lookup table rule. The following parameters can be set:

Id	obligatory and unique for the control group
Name	optional and not necessarily unique
TimeLag	Time lag always given in seconds
Extrapolation	extrapolation of table. Always constant
Interpolation	interpolation of table (constant or linear)
Table	Table with relation between control parameter and controlled parameter

The user can specify a time lag to be applied to the control parameter. If the time lag is greater than zero, not the current value, but values from previous time steps are used. For example, if the time lag is 1 day (86 400 s), the value of the control parameter is taken from the day before, and not the value that corresponds to the current time step.

If a time lag different to zero is applied, care must be taken for the initial phase of the simulation. Until a simulation period equal to the time lag is computed, no input data is available for the Lookup table rule. So the rule gives no output and no controlled parameter is transferred to the structure connected to the rule. Consequently, the structure is considered to be not controlled by D-RTC. For the simulation the parameter value specified for the structure in the D-Flow 1D model is used in this case. Hence, for modeling studies where a time lag for a Lookup table rule is specified the user either has to

◇ take into account the lack of input data for control in the initial phase in the analysis of results or
◇ use initial conditions from restart (a state saved previously; see D-Flow 1D, User Manual.

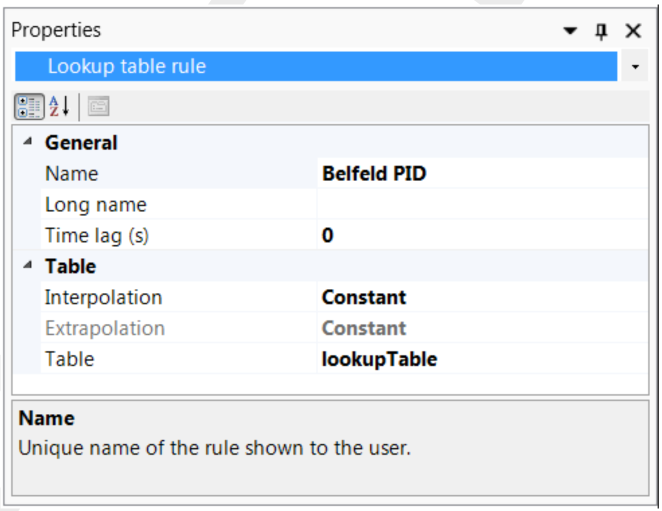


Figure 4.12: *A Lookup table rule in the* ***Properties*** *window*

4.3.2 Time rule

The time rule is the simplest rule; the controlled parameter is defined explicitly as a function of time. The time rule is therefore the only rule without input data from a control parameter. Figure 4.13 shows an example of the use of the time rule in the flowchart.

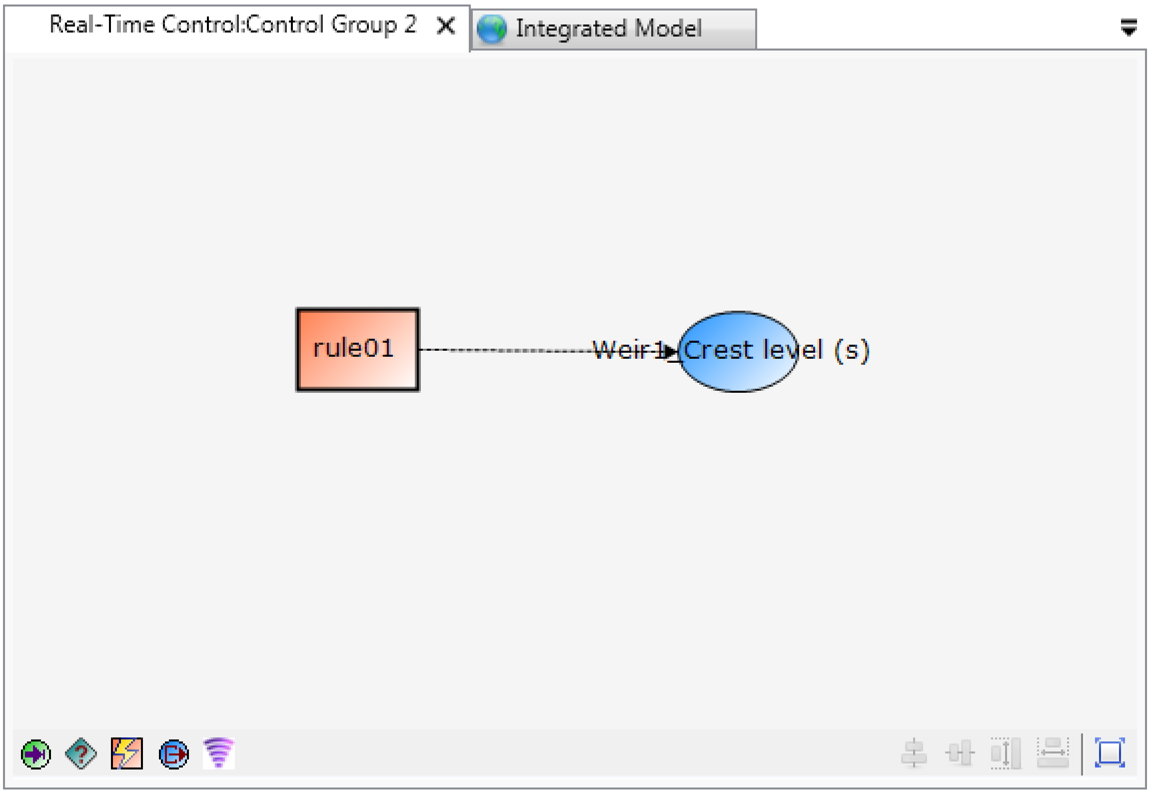


Figure 4.13: A time rule in the flowchart

Figure 4.14 shows the time rule in the **Properties** window. The time rule has the following parameters:

- Timeseries: timetable of the controlled parameter as a function of time
- Interpolation: interpolation within the time table (linear or constant)
- Periodicity: extrapolation of time table. The options are
  - Constant: the last value is maintained
  - Periodic: the table is repeated (both before and after the times in the table)
- Id: obligatory and unique for the control group
- Name: not obligatory and not necessarily unique

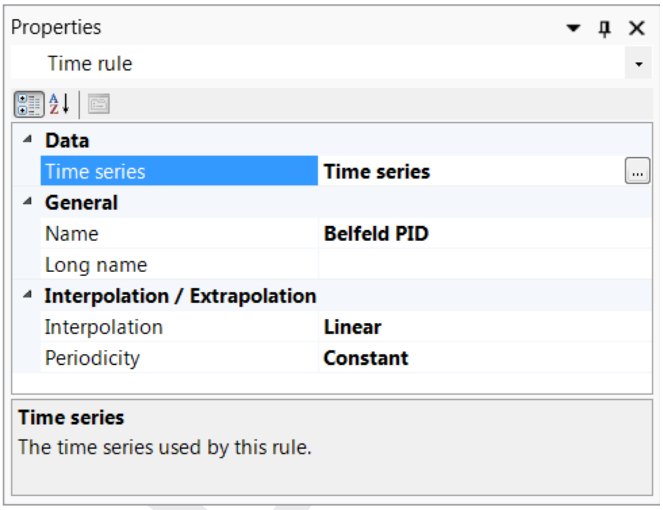


Figure 4.14: *A time rule in the* ***Properties*** *window*

4.3.3 PID rule

Warning:

◇ Previous versions of SOBEK3 and SOBEK2
This PID rule will behave differently from the PID controller in previous releases of SOBEK 3.7.4 and SOBEK 2. Therefor, after opening an old SOBEK model we recommend to calibrate the PID rule again to obtain reliable results.

4.3.3.1 Introduction

The PID (Proportional Integral Derivative) rule is a control loop feedback mechanism used to operate a structure in such way that a specified hydraulic control parameter (e.g. water level or discharge) is maintained. The control parameter can be the water level or the discharge at a specified observation point in the network.

The PID-rule can been seen as a external force on the system using three tuning parameters and read:

$$f(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}, \tag{4.1}$$

where

K_p	gain factor proportional to $e(t)$,
K_i	gain factor proportional to the time integral of $e(t)$ and
K_d	gain factor proportional to the time derivative of $e(t)$.

with $e(t)$ the error (deviation) from the setpoint ($e(t) = x_{sp} - x(t)$). These gain factors must be adjusted for each situation. By adjusting the values the user puts the emphasis of the rule on current deviations from the setpoint of the control parameter (K_p), previous deviations (K_d) and all previous deviations (K_i). This allows the user to set the behavior of the rule such that the structure responds fast or that the response is dampened by previous events. Wheras the interval rule can become unstable with many fluctuations in the controlled parameter, by optimizing the factors the PID rule can be stabilized.

To prevent the whole history of deviations the PID-rule Equation (4.1) is linearized and read in discrete form:

$$f^n = f^{n-1} + K_p \left(e^n - e^{n-1}\right) + K_i\, \Delta t_n\, e^n + K_d \frac{e^n - 2e^{n-1} + e^{n-2}}{\Delta t} \tag{4.2}$$

If necessary, the new value of the control parameter $f(t)$ is adjusted to fit within the physical limits of the structure (i.e. Minimum, MaxSpeed, Maximum).

Gain factors can be positive or negative. The choice of the sign depends on the type of the control structure (e.g. crest level, crest width or gate lower edge level) and the location and type of the hydraulic parameter (e.g. water level or discharge) that is controlled by the PID rule. For example, consider a PID rule that at a bifurcation tries to maintain a constant discharge flowing into one branch by manipulating the crest level of a River weir located in the branch that should receive this constant discharge. In case the discharge flowing into the branch of which its discharge is controlled is too large, this means that the deviation e in the equation above is negative (i.e. the deviation from the actual discharge < 0). From a hydraulic point of view the crest level ($f(t)$) of the river weir is to be raised in order to reduce the discharge flowing into the controlled branch. In order to achieve this, the K_p gain factor should be negative.

4.3.3.2 PID rules in D-RTC

Figure 4.15 shows an example of the use of a PID rule in the flowchart.

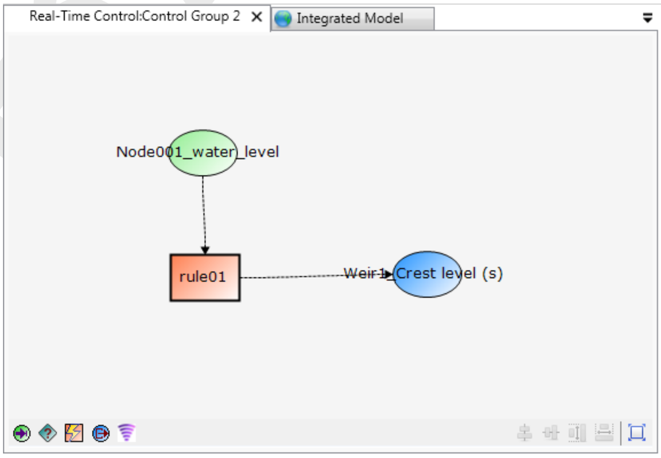


Figure 4.15: *A PID rule in the flowchart*

Figure 4.16 shows the PID rule in the **Properties** window. The PID rule uses the following parameters:

- Setpoint: setpoint of control parameter
  - IsUsingConstantSetpoint: true if the setpoint is constant in time, false if the setpoint is a function of time
  - ConstantSetpoint: value of the setpoint if IsUsingConstantSetpoint is true
  - Table: table with setpoints as a function of time if IsUsingConstantSetpoint is false
  - TableExtrapolation/Interpolation: Linear or block interpolation and constant or periodic extrapolation
- Gain factors: K_p, K_i and K_d.
- Limits: Physical limits of the structure

    - Minimum: minimum value of the controlled parameter
    - Maximum: maximum value of the controlled parameter
    - MaxSpeed: maximum velocity with which the controlled parameter is adjusted
- Id: obligatory and unique for the controlgroup
- Name: not obligatory and not necessarily unique

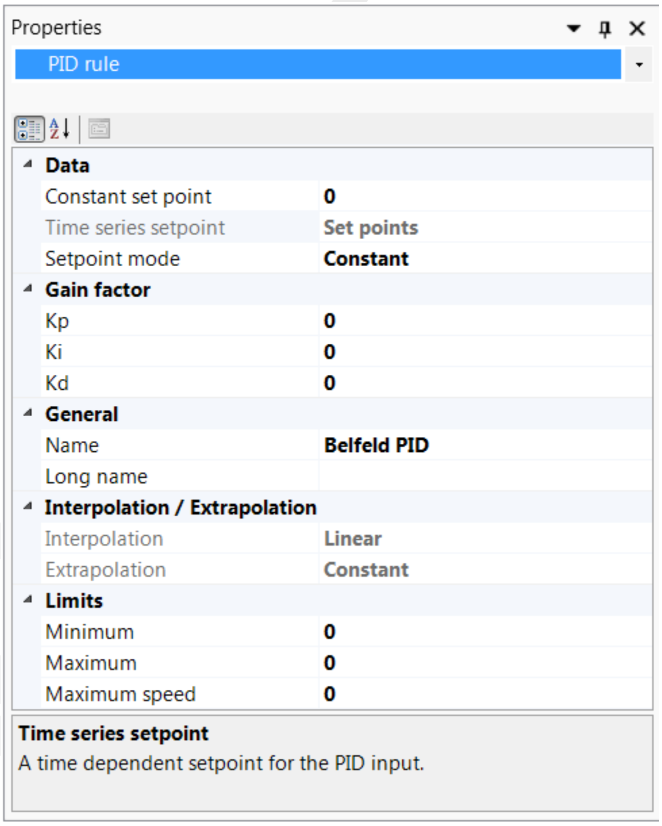


Figure 4.16: *A PID rule in the* ***Properties*** *window*

4.3.3.3 PID rule calibration

The gain factors K_i, K_p, K_d must be calibrated for optimal performance of the PID rule. For example the calibration can be carried out as follows:

- Take K_p, K_d equal to zero, and increase the value of K_i gradually from a small value until the solution starts to oscillate. The sign of K_i must be chosen dependent of the type of structure and the chosen control parameter (see section 4.3.3.4)
- Next divide the resulting value of K_i in half and start increasing K_p with a factor times K_i. Please note that K_p has the same sign as K_i. Again the value of K_p is increased until oscillations appear. K_d remains equal to zero.
- Finally increase the value of K_d (sign of K_d may be opposite of sign of K_i).

A strict procedure for this calibration cannot be presented, since the procedures and results are dependent on the type of model.

4.3.3.4 PID parameter settings

Sign of K_p

The sign of the K_p determines the PID rule output. The error e is defined as the difference from setpoint and system state (see Equation (6.17)). K_p must have a sign such that the PID controller output matches to the intended physical meaning.

Example: If the water level is higher than the target value (the setpoint), the error takes negative values. If the PID rule controls a pump which is configured in such a way that it pumps away excess water with a positive discharge, the sign of K_p must be negative.

Minimum and Maximum

The Minimum and Maximum parameters define constraints for the PID rule output (Equation (6.15)). Because a hydraulic model that is connected to D-RTC can overrule the PID output, the minimum and maximum settings for a PID rule in D-RTC should correspond to the physical properties of the controlled structure in the connected hydraulic model.

Example: A pump structure in a D-Flow 1D model can have a positive discharge only. Consequently, the PID Minimum parameter in D-RTC should be zero and the PID Maximum parameter should be greater zero in order to prevent the hydraulic model from violating the PID logic by overwriting negative PID output with zero. Accordingly, switch on level and switch off level in a D-Flow 1D model should be chosen such that they do not interfere with the PID logic. If the PID controller controls the crest level of a weir, the PID Minimum and Maximum parameter should reflect the minimum and maximum crest level as specified in the hydraulic model.

Initial conditions

The initial conditions have an impact on the performance of the PID controller, because the PID output depends on the PID output from the previous time step (see Equation (6.15)). The initial values for PID output and PID setpoint can be set in a restart file. Restart files can be created and applied with the help of the D-RTC run parameters 'Use restart' and 'Write restart'.

Example: In an integrated model where D-RTC controls a pump structure in D-Flow 1D, the pump capacity from the D-Flow 1D model is transferred to D-RTC as default initial value for the PID output. Indeed pumps are often switched either on or off in practice, so they run either close to full capacity or not, and it makes sense to take the pump capacity as a guess for the initial PID controller output. However, in a model a PID rule is often applied to operate a pump within in the full capacity range, which means that the whole range of discharge values between zero and full capacity is feasible. It can make sense to use a value of 0 as initial controller output here in order to prevent too high pump operations in the first computed time step. The initial PID controller output can be obtained by setting the pump capacity to zero in D-Flow 1D, or by modifying the D-RTC restart file.

4.3.4 Interval rule

The interval rule can be used to operate a structure in such a way that a specified hydraulic parameter is maintained. This controlled parameter can be the water level at a specified observation point in the network, the discharge at a specified observation point in the network.

Figure 4.17 shows an example of the use of an interval rule in the flowchart. An interval rule always needs the input of a control parameter.

Figure 4.18 shows the **Properties** window for an interval rule. There are several parameters available for the interval rule:

- ◇ Setpoints control parameter; this is either a constant set point or a time series. Once a time series has been generated, this is used as set point
- ◇ Interpolation: only used when set points as a function of time are available. The interpolation is between the values in the time-series for the set points. Possibilities are
  - constant
  - linear
- ◇ Below/above limits: Values for controlled parameter when control parameter is above or below the setpoint
- ◇ Deadband: a region in which the interval rule does not respond to deviations in the control parameter from the setpoint
- ◇ Deadband Type: the deadband region can be defined absolute or as a percentage
- ◇ IntervalType: fixed or variable
- ◇ Fixedinterval or Maxspeed: one of the two depending on the IntervalType must be entered

When the interval type is set to fixed, the controlled parameter is adjusted with a fixed amount each timestep. This fixed amount is the parameter FixedInterval. In this mode the value of the controlled parameter is independent of the actual timestep. If the controlled parameter is crest level of a weir and the FixedInterval is set to 1, the crest level will be adjusted with 1 meter every timestep (within the limits of the structure set by the values Below and Above), regardless whether the timestep is a minute or an hour. It is up to the user to set an appropriate value for the FixedInterval.

When the interval type is set to variable, the controlled parameter is adjusted with a velocity, specified by the parameter Maxspeed. This velocity is a maximum velocity. D-RTC checks whether within that timestep the limits of the structure are reached. If so, the actual adjustment is smaller and hence also the actual velocity. In this mode, the actual adjustment of the structure is a function of the timestep. If the timestep is twice as long, the adjustment will be twice as large (within the limits of the structure).

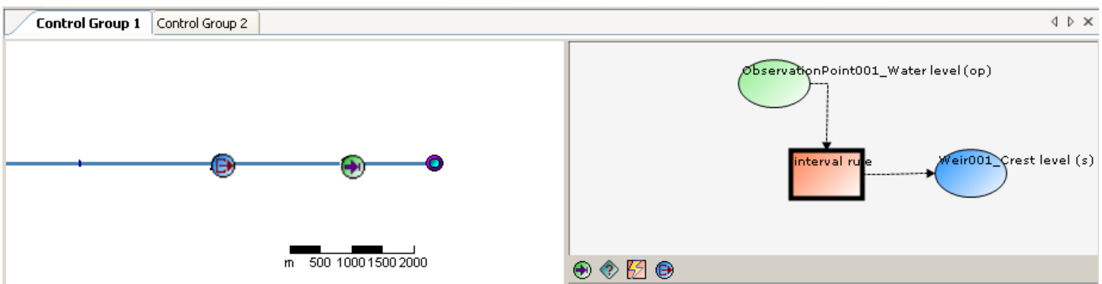


Figure 4.17: *An interval rule in the flowchart*

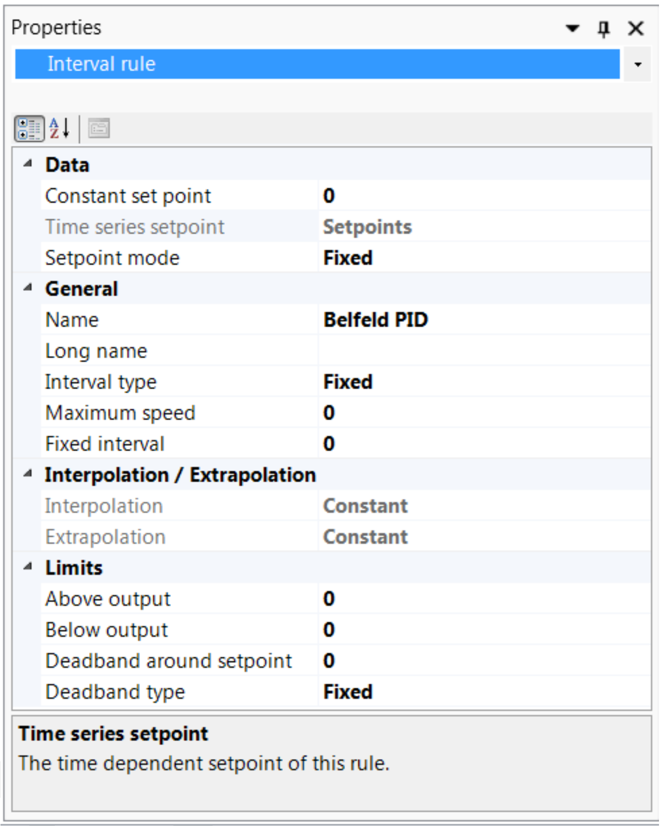


Figure 4.18: *An interval rule in the* ***Properties*** *window*

4.3.5 Relative from time/value rule

The relative time rule can be used to specify the controlled parameter as a function of time, where the time (in seconds) is given relative to the moment that the rule is activated for the first time by a condition. When the rule is activated for the first time, the relative time table is made absolute (= computational time + relative time), thereafter the rule starts at the top of the table and continues downward until the rule is deactivated by a condition. The rule table will remain absolute during the user-defined so called Start period. In case the rule is activated after this start period has passed, the table will be made absolute again. Start period $= 0$, means that the table is made absolute each and every time that the rule is activated. In case the user defined value for $d(\text{value})/dt$ is too small to allow for the in the Table defined changes in control parameter, D-RTC will divert from these defined parameter values in such way as to best fit the overall table. $d(\text{value})/dt = 0$, means that there is no restriction in change in parameter over one time step. When it reaches the end of the table, the value of the controlled parameter is kept constant at the last value.

The relative from value rule is similar, except that the table is started not at the top, but at the value of the controlled parameter.

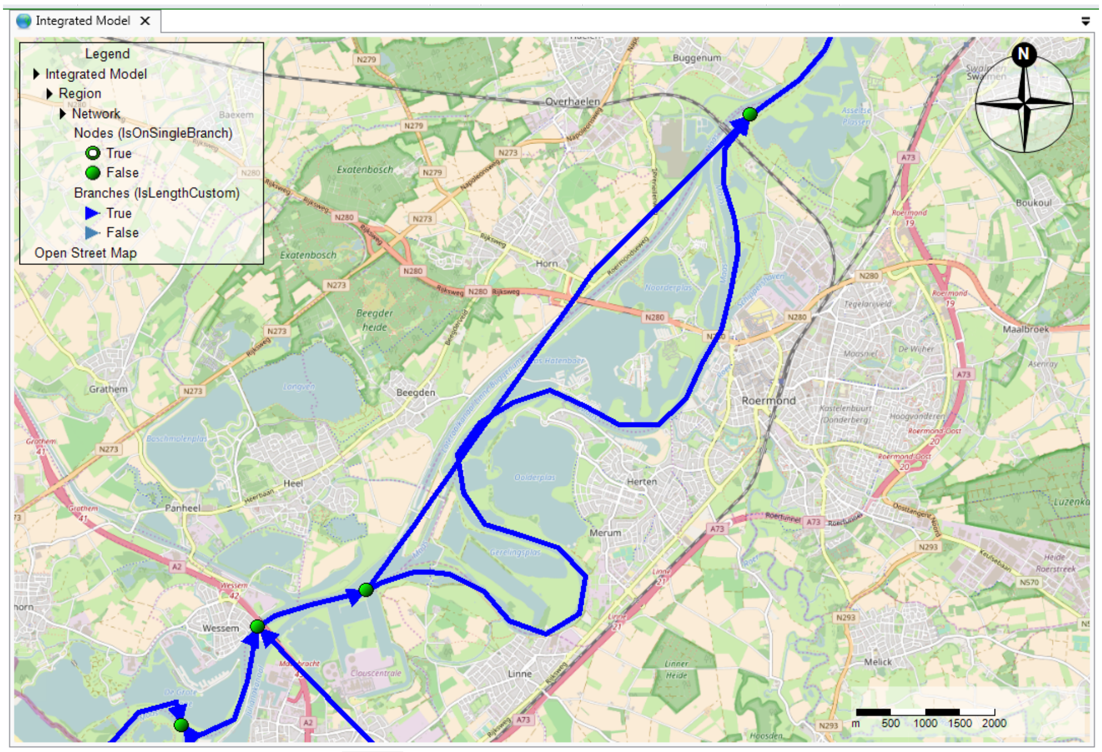


Figure 4.19: *D-Flow 1D model of the River Meuse with close-up for the "Maasplassen" region (Roermond, the Netherlands); background map: http://openstreetmap.org*

4.3.6 Invertor rule

The output of the invertor rule is the input with a changed sign. Invertor rules can be used to model hydraulic bypasses in a river network by connecting lateral sources. An application example are the "Maasplassen" near Roermond in the Netherlands. Maasplassen are gravel pits that are hydraulically connected to the Meuse river. The name "Maasplassen" is also used for the region that is shaped by the gravel pits (Becker *et al.*, 2011). Figure 4.19 shows a map with a section of a D-Flow 1D model for the "Maasplassen".

During high water the gravel pits are flooded by the river Meuse. Hydraulically the pits act like short cuts in the meandering course of the river, which reduces the travel time of the flood wave. In the model for the river Meuse this bypass effect has been accounted for by introducing two lateral sources into the hydraulic that are virtually connected with the help of the invertor rule (Figure 4.20). The input location of the invertor rule is the lateral source that represents the upstream end of the bypass (ID 317), and the output location is set to 318, which is the lateral source that represents the downstream end of the virtual bypass.

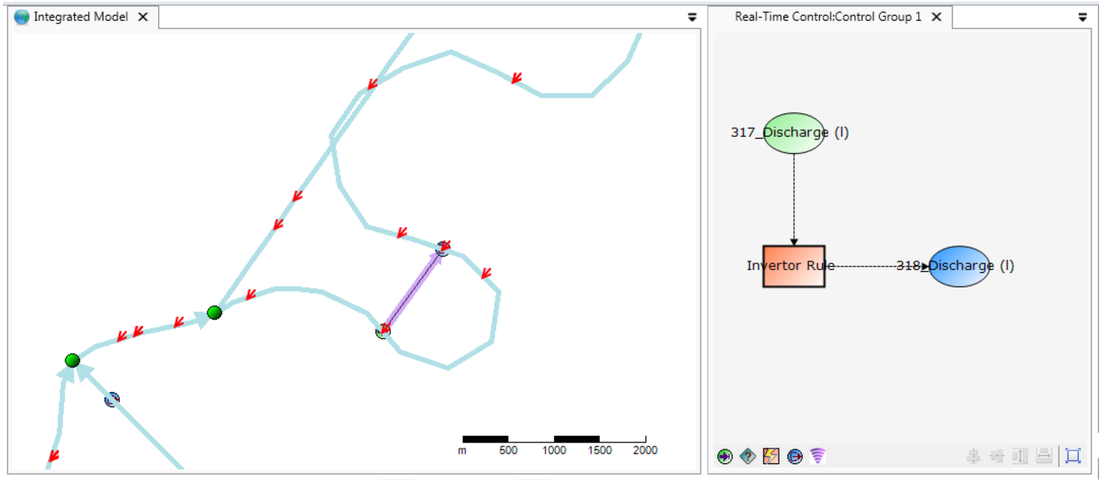


Figure 4.20: *Two lateral sources connected with an invertor rule*

The lateral source at the upstream end of the bypass with ID 317 has a waterlevel–discharge relation. Beginning with an elevation of 20.58 m, water is extracted from the river course in dependence of the water level. Lateral source 318 has a constant discharge value of 0. This value is overwritten by the invertor rule, where the lateral source 318 is set as output location. Technically, data from the input location of the invertor rule is passed with changed sign to the receiving lateral source in the next time step.

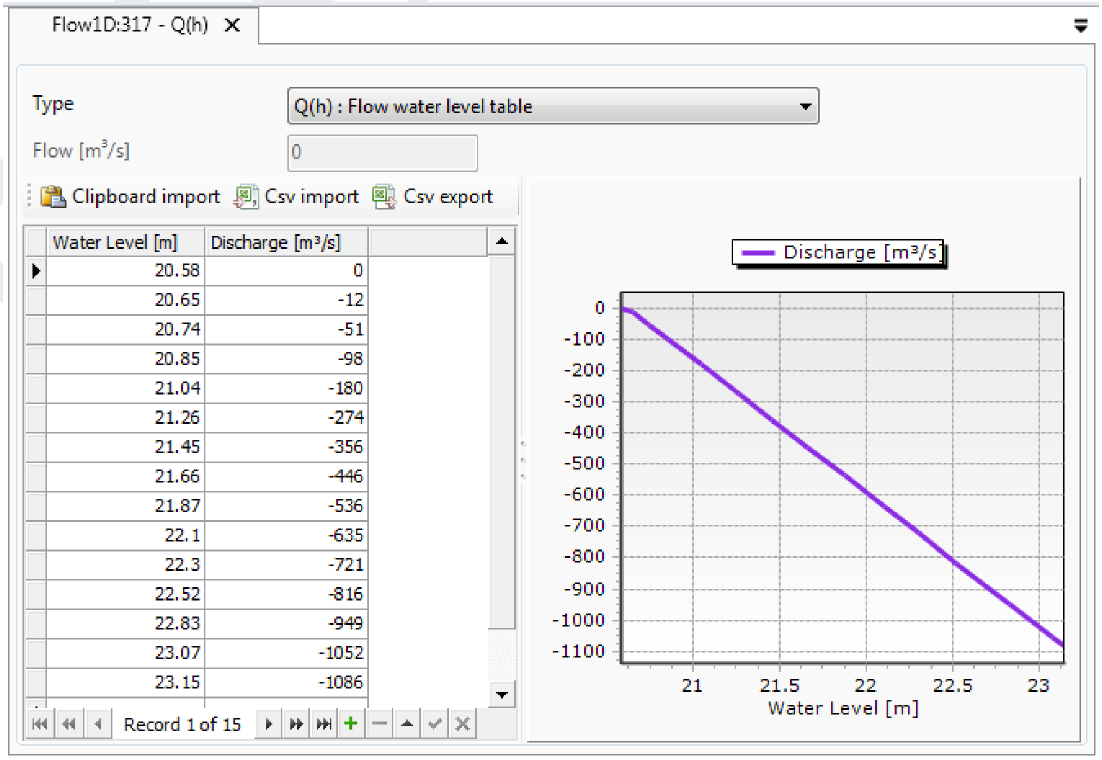


Figure 4.21: *Table and graph for the relation between water level and discharge for the lateral source that represents the upstream end of the bypass*

4.4 Output locations

For all controlflows there is an output location needed. Selecting the output locations is supported by the GUI. First select the input icon ( ) and then click some where in the flow chart. In the flowchart an object will appear with the text "[Not Set]" in it. Press the right mouse button and select the output location. This can be done via several drop down list or via a table option, see Figure 4.22 or Figure 4.25.

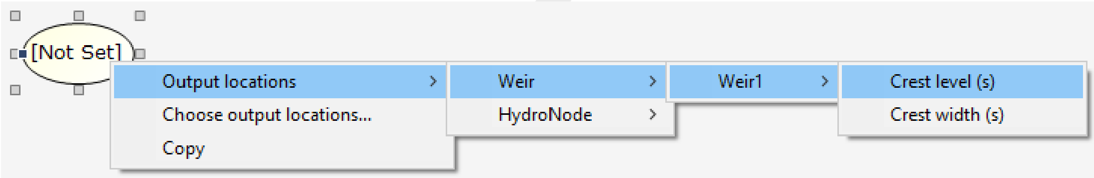


Figure 4.22: *Select output location via drop down lists*

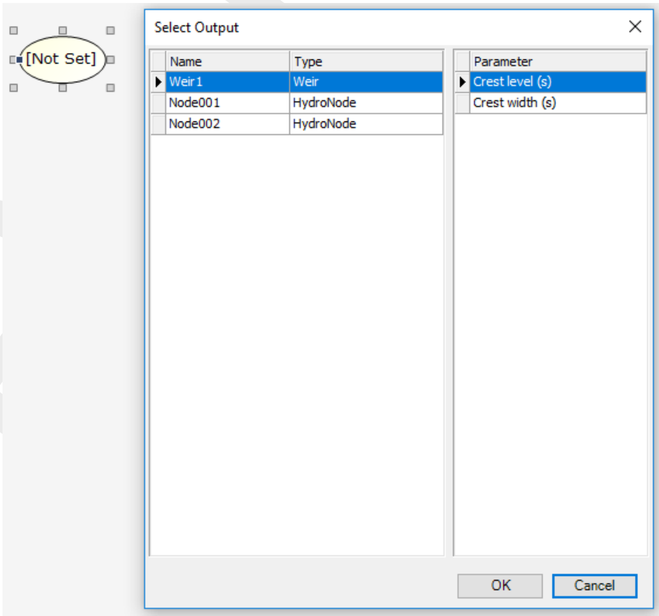


Figure 4.23: *Select output location via a table*

Figure 4.24: *Selected output location, as given in the object on the flowchart.*

4.5 Signals

With signals you can set the setpoints of the PID-rule or an Interval-rule dependent on the hydraulic computation via a predefined table. An example of such table is given in Table 4.2

Table 4.2: *Signal table for the setpoint (ex. crest level [m])*

Water level	Setpoint
1	-2.5
2	-3.5
5	-4.5
10	-5.5

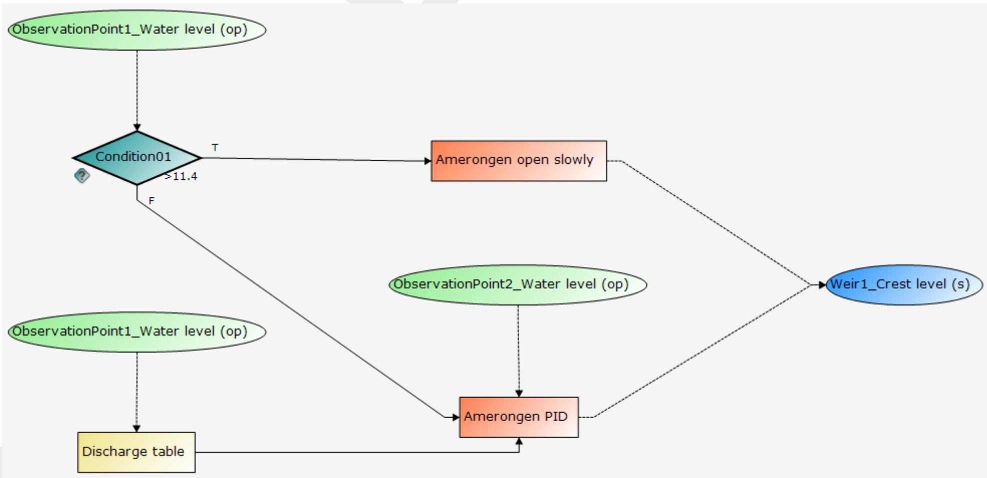


Figure 4.25: *Example flow chart layout for a signal, adjusting the setpoint for a PID-rule*

5 Simulation and model output

The simulation results of D-RTC can be accessed as described in section 3.3.4. D-RTC Simulation results are also written into a temporary directory of the user's local settings:

```
c:\Documents and Settings\<user>\Local Settings\Temp\
```

where <user> is a placeholder for the user's name.

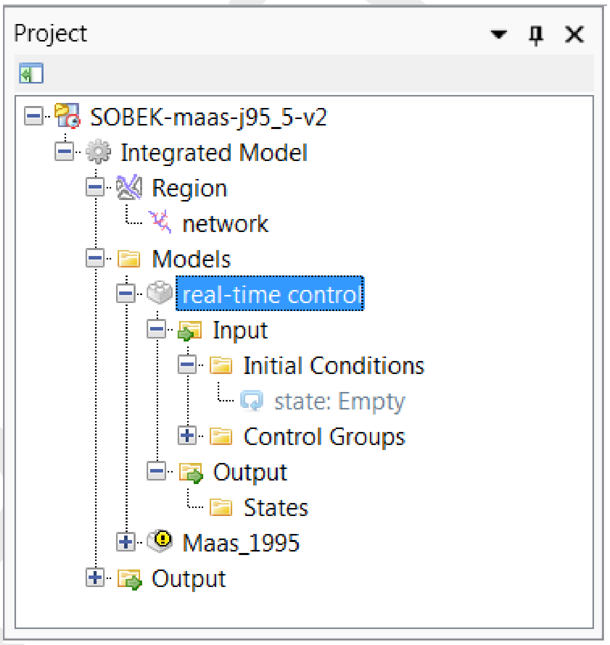


***Figure 5.1:** D-RTC-model selected in the **Project** window*

Right-click in the **Project** window and choose *Open last working directory* to navigate to the current working directory where the simulation input and output is stored. The file <timeseries_0000.csv> contains the time series of all model objects of the D-RTC model as comma-separated value table. This file can be easily opened and postprocessed with text editors or programs like Microsoft Excel or Matlab in order to analyze the simulated values related to input and output locations and the status of D-RTC objects coherently.

Furthermore, the following files can be found in the temporary directory:

- <rtcDataConfig.xml>
- <tcRuntimeConfig.xml>
- <rtcToolsConfig.xml>
- <state_import.xml>
- <statePI.xml>
- <timeseries_export.xml>

With this set of xml-files a complete RTC-Tools model is given. RTC-Tools is the computational core of D-RTC and can be considered as the research version of D-RTC (see http://oss.deltares.nl/web/rtc-tools for details). <diag.xml> and <state_export.xml> are RTC-Tools output files.

6 Technical reference

6.1 Overview

D-RTC is always coupled to an hydraulic model, such as D-Flow 1D, D-Flow FM (2D/3D Flexible Mesh) or D-RR (Rainfall Runoff). Hydraulic structures in these models can be controlled by operating rules and controllers in combination with triggers. Consequently, D-RTC distinguishes basically two layers: (1) triggers, and (2) operating rules. This chapter covers layers (1) and (2), beside general purpose components such as the lookup table which are used in both layers.

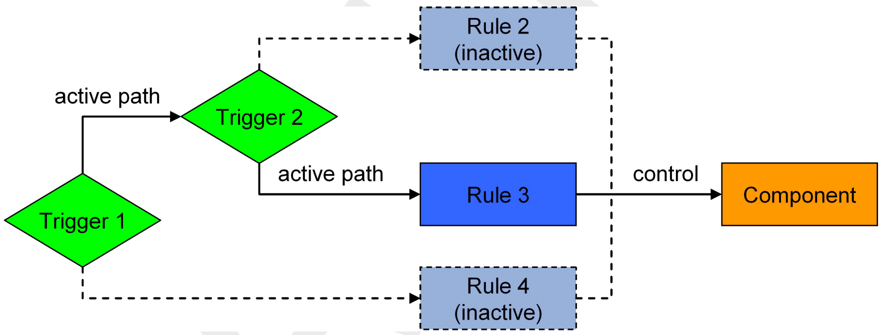


Figure 6.1: *Hierarchical definition of deadBand and standard triggers*

According to our definition, a trigger implements conditions for

◇ defining *when* an operating rule, controller or another trigger is applied,
◇ returning true or false, e. g. if a threshold is crossed or not.

Operating rules and controllers

◇ define *how* a structure operates, and
◇ return a value for a controlled parameter, e.g. a gate opening or pump discharge, which is picked up by the hydraulic model the D-RTC model is coupled to.

A combination of triggers with operating rules and controllers forms a binary decision trees such as given in Figure 2.2. Triggers may connect to other triggers. This feature makes it possible to build complex decision trees, such that a hydraulic structure is controlled with different rules for each case, respectively (Figure 6.1). Decision trees must be constructed such that there is only one active path at a time towards a hydraulic structure. Rules and triggers that are connected to an output path of a trigger (true or false) that is not active in the current time step are not evaluated.

From a mathematical point of view, all features in this chapter compute their outputs from available data either from the previous time step $k-1$ (explicit) or from output of previous components of the same time step k (implicit).

Note: Triggers are always evaluated before rules.

6.2 General purpose components

6.2.1 Accumulation

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file..

The component adds an input x to the state y. The equation of the "accumulation" components reads

$$y^k = y^{k-1} + x^k \tag{6.1}$$

where $k-1$ refers to the previous time step and k to the current.

This component is used to get the accumulation of a value over the simulation period.

6.2.2 Mathematical expression

The expression consists of a mathematical equation of the form

$$y^k = x_1^{k-1 \vee k} + x_2^{k-1 \vee k} \tag{6.2}$$

The following operators are supported:

$+$	summation
$-$	subtraction
$*$	multiplication
$/$	division
$\min$	minimum (case sensitive, thus $\min$(B-A))
$\max$	maximum.(case sensitive, thus $\max$(B-A))
pow	Power function, pow(A,B) $= A^B$ (case sensitive). **This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.**.

```xml
<trigger>
   <expression id="G*H">
      <x1Series ref="IMPLICIT">[Input]Point_1/depth</x1Series>
      <mathematicalOperator>*</mathematicalOperator>
      <x2Series ref="IMPLICIT">9.81</x2Series>
      <y>G*H</y>
   </expression>
</trigger>
<trigger>
   <expression id="SQRT(G*H)">
      <x1Series ref="IMPLICIT">G*H</x1Series>
      <mathematicalOperator>^</mathematicalOperator>
      <x2Series ref="IMPLICIT">0.5</x2Series>
      <y>celerity</y>
   </expression>
</trigger>
```

The recursive use of expressions (another expression as one of the two terms or for both) enables the implementation of more complex mathematical expressions (check the example in the configuration section).

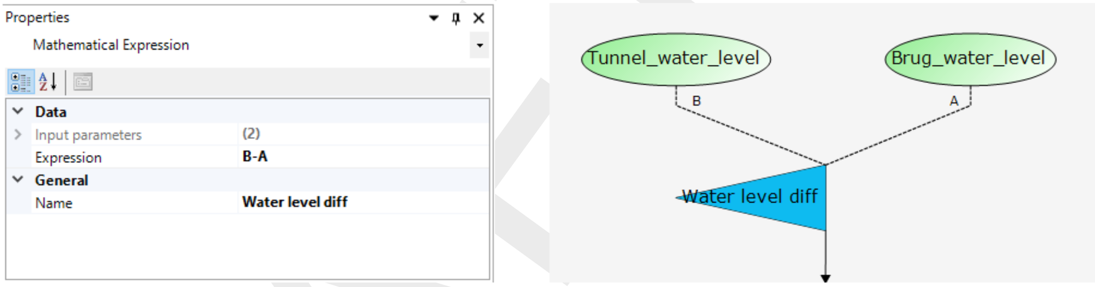
Expressions are used for a wide range of applications. A very common example is the computation of a difference in water level upstream and downstream of a barrier. This difference is used as input for a trigger within a decision tree.

The reference to the time step $k-1 \vee k$ (previous or current) depends on whether the time reference in the time series definition is defined as implicit or explicit (see section 6.1 and C.4).

An example of the mathematical expression

$$f = B - A \tag{6.3}$$

is given in Figure 6.2. Where B represent the water level at the loccation 'Tunnel' and A represent the water level at the location 'Brug' (i.e. bridge).



***(a)** Properties panel* ***(b)** Graphic view*

***Figure 6.2:** Layout for a mathematical expression (difference between water levels)*

6.2.3 Gradient

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The governing equation of the gradient reads:

$$y^k = \frac{x^k - x^{k-1}}{\Delta t} \tag{6.4}$$

6.2.4 LookupTable

The rule supplies a piecewise linear 1D lookup table according to

$$y^k = f(x^{k-1 \vee k}) \tag{6.5}$$

The lookup table rule (section 4.3.1) uses this feature. This rule is a simpler version of the dateLookupTable (section 6.3.2).

6.2.5 Lookup2DTable

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The rule supplies a piecewise linear 2D lookup table according to

$$y^k = f(x_1^{k-1 \vee k}, x_2^{k-1 \vee k}) \tag{6.6}$$

6.2.6 MergerSplitter

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The merger rule provide a simple data hierarchy by choosing the output y equal to the first of several input values $x_1, x_2, ..., x_n$ which is non-missing. Furthermore, additional output includes the sum of all input values.

6.2.7 UnitDelay

The unit delay operator is an auxiliary tool for making data from time steps prior to the previous time step available in the simulation. By using this operator, we can refer to a historical release, for example in an operating rule, without abandoning the restarting features of the model based on the system state of a single time step. It reads

$$y^{k+1} = x^k. \tag{6.7}$$

6.3 Operating rules and controllers

6.3.1 Constant

Functional principle

This simple rule defines a user-defined constant output y^k according to

$$y^k = \text{const}. \tag{6.8}$$

Application

A constant rule is typically applied in combination with triggers (see Section 6.4) for on-off control (Åström and Hägglund, 1995), e.g. a weir fully opened or fully closed. The true and the false output of a trigger are each connected with one constant rule.

6.3.2 DateLookupTable

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The date lookup table is a 2D lookup table with the time axis as one of its dimensions. Its discrete form reads

$$y^k = f(t, x^{k-1 \vee k}) \tag{6.9}$$

The resolution of the time axis t is in days of the year. The value axis x may have any range. A typical application of the rule would be the definition of a minimum release of a reservoir as a function of the day of the year and the water level of the reservoir.

6.3.3 DeadBandValue

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The dead band value rule is a discrete rule for suppressing the output of another rule until its rate of change becomes higher than a certain threshold. It reads

$$x^k = \begin{cases} x^{k-1} & \text{if } \left|x^k - x^{k-1}\right| < \text{threshold} \\ x^k & \text{otherwise} \end{cases} \tag{6.10}$$

It is often applied to limit the number of adjustments to movable elements of hydraulic structures in order to increase their life time.

6.3.4 GuideBand

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The guide band rule provides a linear interpolation from input x to output y, if x is between two input threshold $x_{\min}$ and $x_{\max}$. Otherwise, the output is limited to defined minimum and maximum output threshold $y_{\min}$ and $y^{\max}$. The rule reads

$$y^k = \begin{cases} y_{\min} & \text{if } x^{k-1 \vee k} \le x_{\min} \\ y_{\min} + \frac{x^{k-1 \vee k}(y_{\max} - y_{\min})}{x_{\max} - x_{\min}}, & \text{if } x_{\min} < x^{k-1 \vee k} < x_{\max} \\ y_{\max} & \text{if } x_{\max} \le x^{k-1 \vee k} \end{cases} \tag{6.11}$$

A graphical representation of the rule is presented in Figure 6.3.

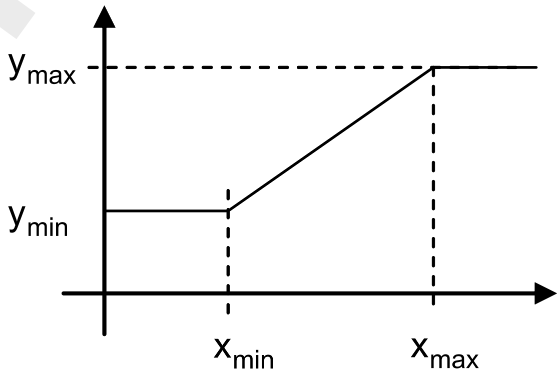


Figure 6.3: *Graphical representation of guideBand rule*

The input and output thresholds $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ can be constant, a function of time or provided by a time series, e.g. from an external input or a result from the execution of a prior rule.

A typical application of the rule is the enforcement of a reservoir storage s between a certain range and the use of the available storage for equalizing the release. If the storage is approaching or down-crossing the lower storage limit $s_{\min}$, the release is set to the minimum flow (zero is no minimum flow is defined). If the storage is approaching or up-crossing the upper limit $s_{\max}$, the release is set to maximum capacity.

6.3.5 Interval

The interval controller is a simple feedback controller according to the control law

$$y^k = \begin{cases} y_{\max} & \text{if } x^{k-1} > sp^k + \frac{1}{2}D \\ y_{\min} & \text{if } x^{k-1} < sp^k - \frac{1}{2}D \\ y^{k-1} & \text{otherwise} \end{cases} \tag{6.12}$$

where x^{k-1} is an input variable, sp^k is a setpoint, D is a dead band around the setpoint, and y^k is the controller output.

6.3.6 Limiter

In contrary to the deadBand rule defined above, the discrete limiter rule restricts the change of a variable to a relative threshold p. It reads

$$y^k = \begin{cases} (1-p)x^{k-1} & \text{if } x^k < (1-p)x^{k-1} \\ (1+p)x^{k-1} & \text{if } x^k > (1+p)x^{k-1} \\ x^k & \text{otherwise} \end{cases} \tag{6.13}$$

where p is the maximum relative rate of change. The configuration also accounts for an absolute rate of change according to the condition $x^k < \Delta p\, x^{k-1}$ where Δp is the absolute rate of change.

A typical application of the rule is the limitation of release changes from a reservoir for avoiding too steep flow gradients downstream.

6.3.7 PID rule

6.3.7.1 Functional principle

The Proportional-Integral-Derivative controller (PID controller) is a generic feedback controller including an optional disturbance term commonly used in industrial control systems (see Åström and Hägglund, 1995, for more information). It reads

$$\begin{aligned} e(t) &= x_{\text{sp}}(t) - x(t) \\ y(t) &= K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \end{aligned} \tag{6.14}$$

where $e(t)$ is the difference between a process variable $x(t)$ and a setpoint $x_{\text{sp}}(t)$, K_p, K_i, K_d are the proportional, integral and derivate gain factors, respectively. $y(t)$ is the controller output.

The proportional gain factor K_p controls according to the deviation of the process variable x to the target value (the setpoint) x_{sp}. Consequently, to some extent the gain factor K_p is a conversion factor between units of the controller output y and the proces variable x. If, for example, a PID controller is used to achieve a certain discharge setpoint in a river by means of controlling the crest level of a weir, K_p will be in the order of magnitude of the ratio between a typical discharge and a typical crest level value. The integral gain takes into account the cumulated historical deviations between setpoint and process variable. The differential gain factor controls according to the current rate of change of deviation between setpoint and process variable.

The discrete form of the linearized Equation (6.14) in D-RTC reads

$$y^k = y^{k-1} + K_p \left(e^k - e^{k-1}\right) + K_i \, \Delta t \, e^k + K_d \frac{e^k - 2e^{k-1} + e^{k-2}}{\Delta t} \tag{6.15}$$

$$y_{\min} \leq y^k \leq y_{\max} \tag{6.16}$$

with the user-defined lower and upper bounds of the controller output $y_{\min}$ and $y_{\max}$, where

$$e^k = sp^k - x^{k-1} \tag{6.17}$$

The implementation allows to limit the rate of change to a user-defined maximum velocity $v_{\max}$. The lower and upper bounds are modified as follows:

$$y^k_{\max} = \min\left(y^{k-1} + \Delta t \, v_{\max}, y_{\max}\right) \tag{6.18}$$

$$y^k_{\min} = \max\left(y^{k-1} - \Delta t \, v_{\max}, y_{\min}\right) \tag{6.19}$$

6.3.7.2 Application

The PID controller is used to achieve strive or target values (e.g. water levels) in a water system.

PID controllers can also be applied for the calibration of hydraulic models with hydraulic structures when the control parameters are unknown, but water levels in the vicinity of the structure (mostly upstream of weirs). The approach is to use a measured water level as setpoint. Calibration parameter is the roughness value of the river bed. So the weir is always controlled in such a way that the weir operational mode (i. e. the crest level) is not of influence for the water level. In principle a water level boundary condition is set for the upstream reach. Note that in this case the past aggregation for a PID setpoint time series is suitable.

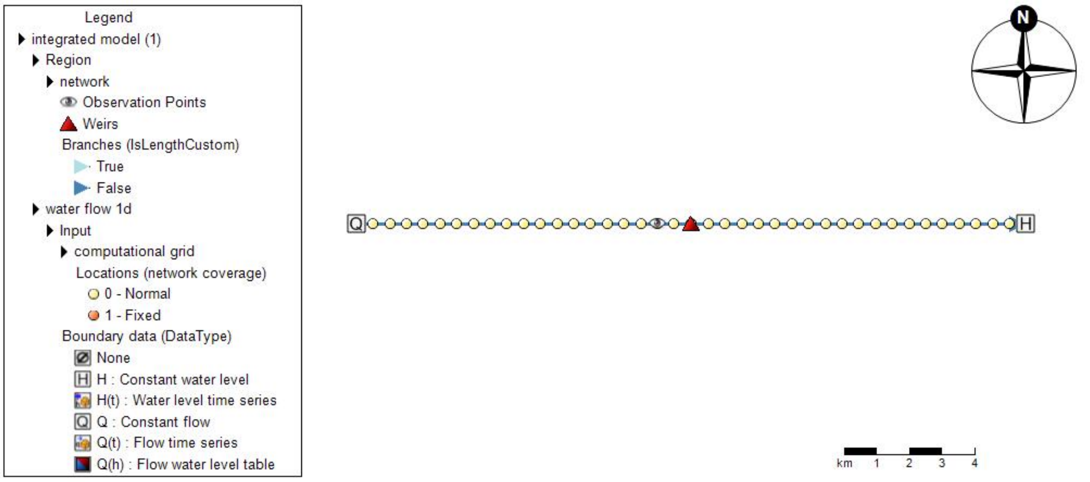
6.3.7.3 Example

The schematization and the computational grid of a simple SOBEK channel flow model are shown in Figure 6.4a. The channel has a constant slope of 0.0002 and a trapezoidal cross section as shown in Figure 6.4b. The simulation time step size is 10 min. The channel has an upstream a discharge boundary condition with a constant value of 5 m^3 s^{-1} and a downsteam water level boundary condition with 2 m. At chainage 10 000 m is a weir, upstream of the weir is an observation point.

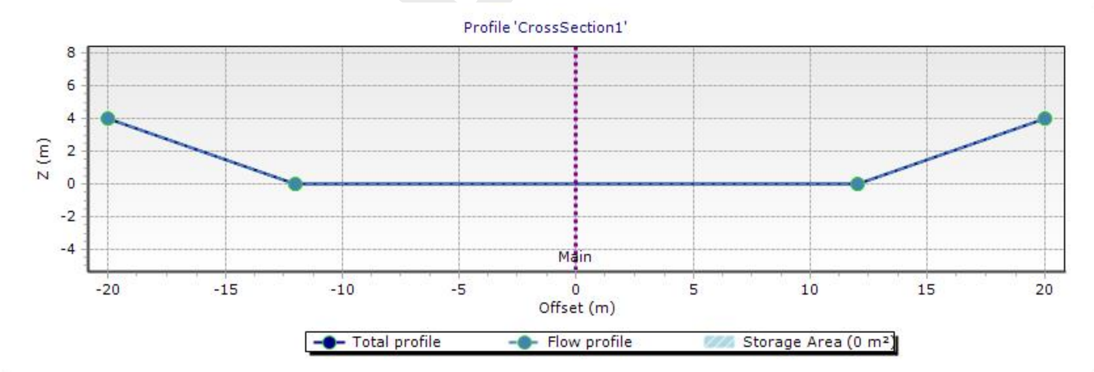
The weir in the channel is controlled with a PID controller rule. The PID controller must control the weir (i. e. move the crest level) in such a way that the water level at the observation point upstream of the weir meets the target values in Table 6.1. Values between the entries of Table 6.1 are to be interpolated. The target values from Table 6.1 are used as setpoint time series for the PID controller, and the controller parameters are $K_p = K_d = 2$ and $K_i = 0$.

Table 6.1: *Target water level for the observation point*

time step k	wall clock time	time series value x in m
1	2000-01-20 00:00:00	4.5
2	2000-01-22 12:00:00	4.5
3	2000-01-22 20:00:00	5.5
4	2000-01-25 00:00:00	5.5



(a) *Schematization and computational grid*



(b) *Cross section*

Figure 6.4: *Simple channel model (SOBEK 3.3)*

Figure 6.6 shows the computed crest level of the weir and the water level at the observation point over simulation time. The results show a very good match between setpoint and computed water level. Figure 6.5 shows the water level along two longitudinal profiles for two time steps and the corresponding crest levels at the weir.

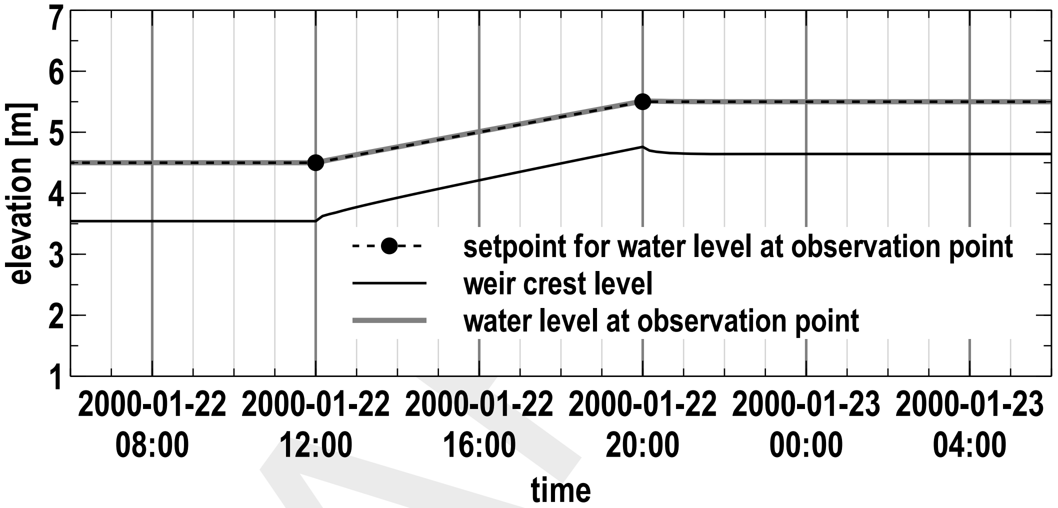


Figure 6.5: *PID controlled crest level of the weir and corresponding water level at the observation point*

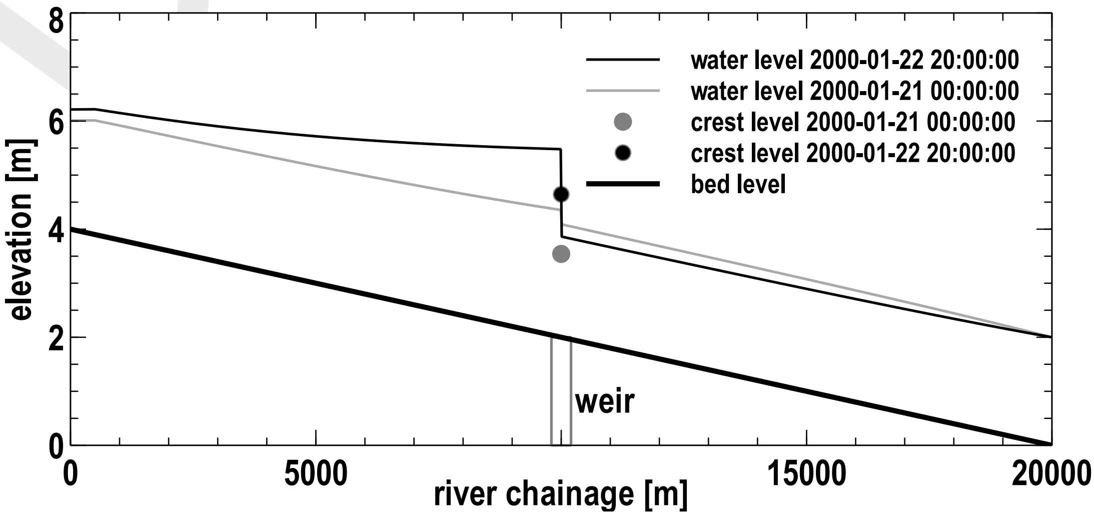


Figure 6.6: *Longitudinal profile of water level for two time steps and crest level of the weir*

6.3.8 The absolute time rule (`timeAbsolute`)

6.3.8.1 Functional principle

A time rule is a time series that defines the state of a control variable. The time series is given as as absolute time stamps in wall clock time. It can be interpreted as a boundary condition for structure parameters. In most cases the future aggregation is desired. This means that a point in time is to be interpreted as “from now on”.

This rule reads

$$y^k = x^k \tag{6.20}$$

where x is an external time series or output of a previous model.

6.3.8.2 Application

Common application cases for the absolute time rule are the following:

- ◇ Calibration of a hydraulic model with structures: the position of crest levels is known as historical value. The historical values are applied as absolute time rule.
- ◇ Hydraulic models in operational systems: control has been determined outside the hydraulic model, and the control settings are applied as time series.

6.3.8.3 Example

An example time series for an absolute time rule is given with Table 6.2. This time series is applied for the weir node in the simple SOBEK model, which is described in section 6.3.7.3.

A close-up of the simulation results and the discrete values from line 2 and 3 6.2 in are shown in the diagram in Figure 6.7. Beginning at 12.00 h the crest level of a weir rises from 2.0 m up to 5.0 m within 8 h. The corresponding time rule output is shown as solid line. The crest rises not before 12.00 h and the movement of the crest stops at 20.00 h.

Table 6.2: *Time rule time series example*

time step k	wall clock time	time series value x in m
1	2000-01-20 00:00:00	2.0
2	2000-01-22 12:00:00	2.0
3	2000-01-22 20:00:00	5.0
4	2000-01-25 00:00:00	5.0

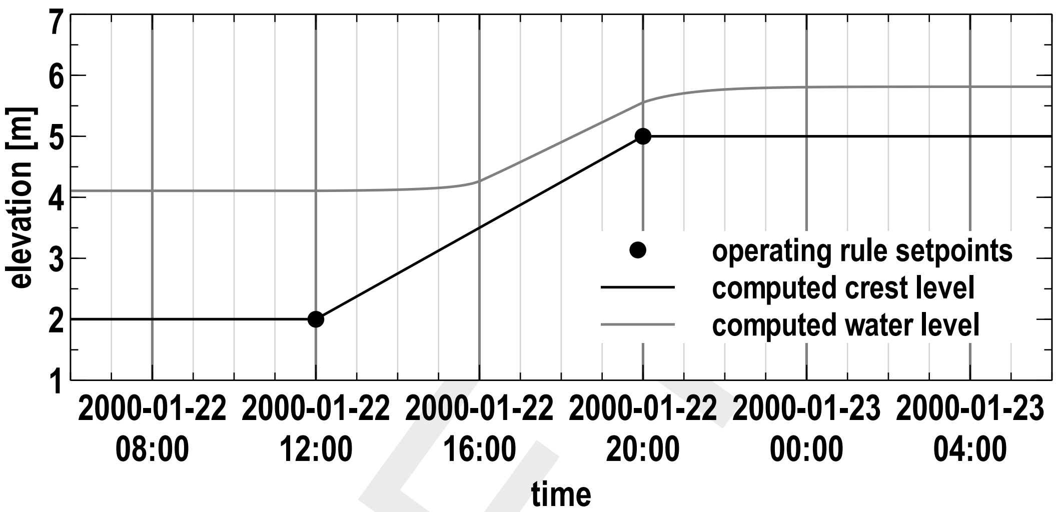


Figure 6.7: *Time rule time series*

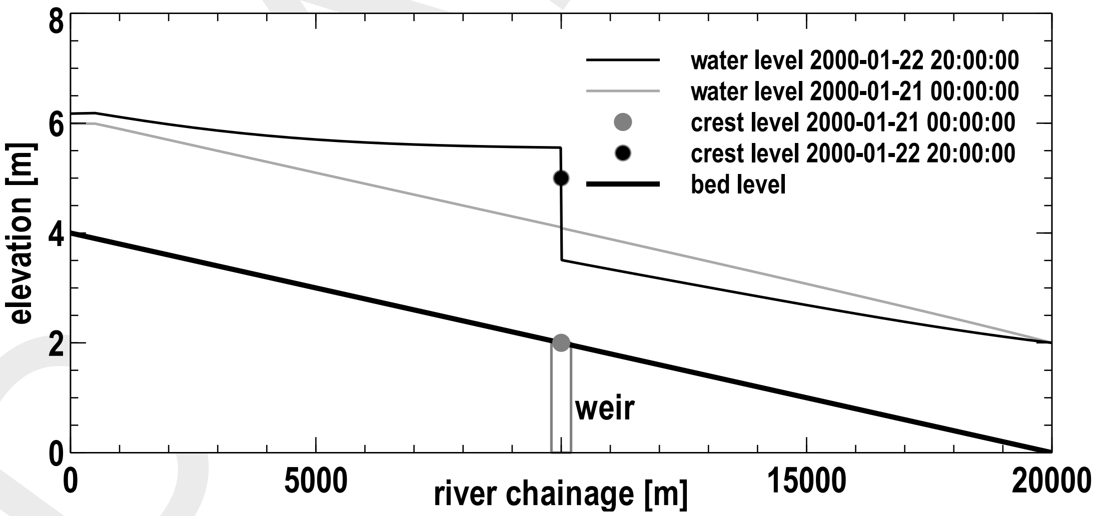


Figure 6.8: *Time rule side-view*

6.3.9 The relative time rule (`timeRelative`)

6.3.9.1 Functional principle

The Relative Time Rule creates a time series for a control variable relative to the current time step. The time series is based on function between relative time steps and the values of the control variable.

This rule reads

$$y^k = x^\tau \tag{6.21}$$

where τ is a relative time reference. When the rule is switched on, the relative time is either

1 put to zero or
2 put to a value based on an existing y for which equation (6.21) is fulfilled.

6.3.9.2 Application

The relative time rule applies for large structures like weirs or gates that move slowly and the opening and closure time must be taken into account for the model. A relative time rule works in combination with hydraulic triggers when the exact points in time of the control time series are not known beforehand.

6.3.9.3 Example

Table 6.3 contains a linear relation between the relative time in seconds and the crest level of a weir. Figure 6.9 shows a graphical representation of Table 6.3. The slope of the curve defines the speed the crest level moves, in the current example this is

$$v_{\text{weir}} = \frac{5-2}{28800}\,\text{m/s} = 1.0416 \times 10^{-4}\,\text{m/s}. \tag{6.22}$$

The relative time rule is related to the current time step. The parameter “FromValue” defines at which point to start when the rule is activated. If the parameter “FromValue” is set to false, the rule starts with the first entry (time reference option “zero” in section 6.3.9), if it is true, the relative time period starts at the position the crest level has at the current time step (second time reference option in section 6.3.9). Figure 6.10 shows the effect of the “FromValue” parameter: before activating the relative time rule at 22-01-2000, 12:00 hours, the crest level is at 2.5 m. If the “FromValue” propertiy is set to false, the crest level goes down to the first entry of the Table 6.3 which is 2.0 m, before it rises to 5.0 m according to Table 6.3. When “FromValue” is set to true, the starting point in relative time that corresponds to the initial crest level of 2.5 m is determined first, and the closure of the weir starts from this value. The total operation of the weir is shorter, because not the full range of the table is used.

Table 6.3: *Relative time rule lookup table for the crest level of a weir*

relative time in s	crest level in m
0	2.0
28800	5.0

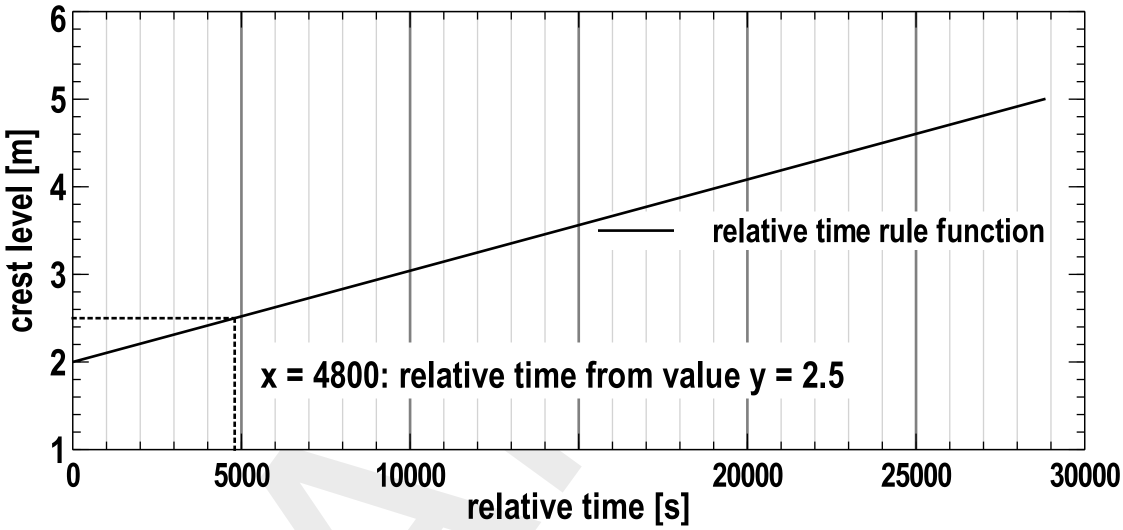


Figure 6.9: *Relative time rule function and relative time from Value 2.5*

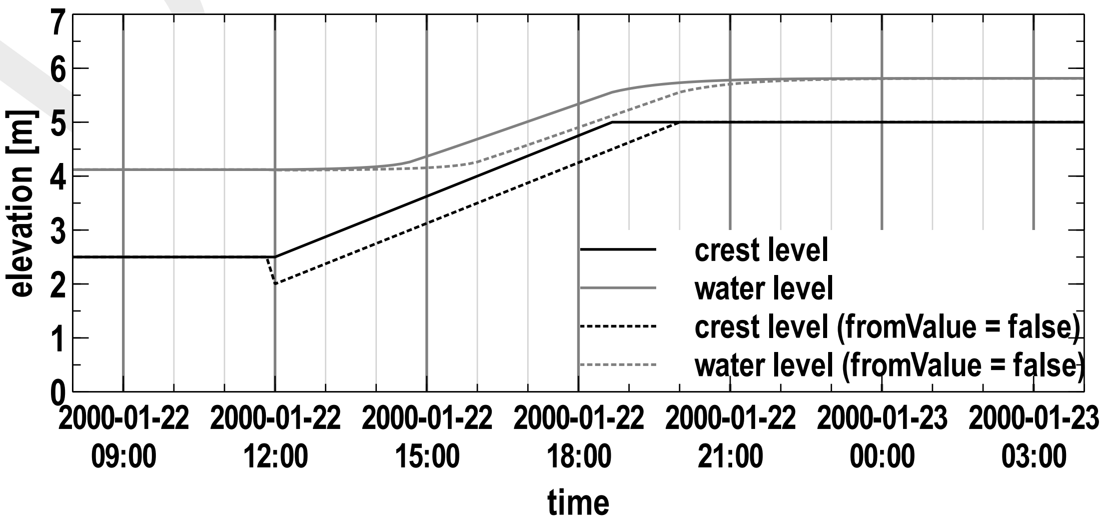


Figure 6.10: *Crest level controlled with relative time rule and "FromValue" parameter true/false and the corresponding water level over time*

6.4 Triggers

6.4.1 Introduction

Originally a trigger is an element of a fire arm that actuates a mechanism to fire a bullet. In electronics the word trigger describes a specific type of an electronic circuit that releases a signal or a switching operation at a certain event. In an analogue way we use the trigger to describe an abstract mechanism that releases (or triggers) another trigger mechanism or a control action, which is then described by an operating rule. A very simple triggering mechanism is an alarm clock signal upon which an operator switches off a pump at a specific point in time. The more advanced triggers in computer models usually need input data.

A trigger output y in RTC-Tools is true or false, which will be represented as $y = 1$ (true) or $y = 0$ (false) in the following. So technically, a trigger in RTC-Tools operates as a switch between two states.

6.4.2 Standard

The standard trigger compares two input values and returns true (1) or false (0):

$$y^k = \begin{cases} 1 & \text{if } x_1^{k-1 \vee k} > x_2^{k-1 \vee k} \\ 0 & \text{otherwise} \end{cases} \tag{6.23}$$

Beside the $>$ operator that is used in Equation (6.23) the following operators are supported: $>, \geq, =, \neq, \leq, <$. The standard trigger is the basis of the D-RTC hydro condition (section 4.2.1).

6.4.3 The time trigger

6.4.3.1 Functional principle

A time trigger models a timer. The trigger output y for time step k is given in a time series x. The time trigger is the basis of the D-RTC time condition (section 4.2.2). An example for such a time series is given in Table 6.4.

Table 6.4: *Time trigger time series example*

time step k	wall clock time	time series value x
1	2000-01-15 00:00:00	false
2	2000-01-16 00:00:00	true
3	2000-01-17 00:00:00	false

When using a time trigger, care must be taken for the aggregation. With future aggregation, the control action that is connected to the trigger follows the trigger with delay with one time step. This will be a desired behaviour for many feedback control applications. If a time lag is not desired, it can make sense to apply past aggregation.

$$y^k = x^{k-1} \quad \text{future aggregation} \tag{6.24}$$

$$y^k = x^k \quad \text{past aggregation} \tag{6.25}$$

6.4.3.2 Application

- ◇ winter/summer operational mode
- ◇ inclusion of external decisions (feedforward control), for example “high water forecasted? (yes/no)”

◇ model calibration when control parameters are known.

6.4.4 deadBandTrigger

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

6.4.4.1 Functional principle

The dead band trigger checks the input data for an upper or lower threshold crossing. The trigger returns true in case of an up-crossing of the upper threshold b_{up}. It is inactive in case of a down crossing of the lower threshold b_{down}. In the range in-between, the trigger keeps its former state. The rule reads

$$y^k = \begin{cases} 1 & \text{if } x_1^{k-1 \vee k} > {b_{\mathrm{up}}}^{k-1 \vee k} \\ 0 & \text{if } x_2^{k-1 \vee k} < {b_{\mathrm{down}}}^{k-1 \vee k} \\ y^{k-1} & \text{otherwise} \end{cases} \tag{6.26}$$

where x_1 and x_2 are time series or constant values. The dead band range given by upper and lower threshold can be constant or time series, too. The following operators are supported: $>, \geq, =, \neq, \leq, <$.

6.4.4.2 Application

Dead band triggers are used to switch on and off pumps or turbines and the dead band ensures that the device is not switched on or off too often.

6.4.4.3 Example

Given the simple SOBEK model, which is described in section 6.3.7.3 under tidal sea level variations, the control task is to control the weir in such a way that it is closed (i.e. the crest level is 6 m) if the head difference at the weir is smaller or equal 0.3 m in order to protect the hinterland from sea flooding and salt water intrusion during high tide. If the head difference is larger than 0.3 m the weir must be open (i.e. the crest level is 2 m) in order to discharge water towards the sea during low tide. The expected result is given in Figure 6.11.

With the help of an expression (see section 6.2.2) the head difference upstream and downstream the weir is computed. The weir is controlled with two constant rules section 6.3.1:

◇ Weir open returns a crest level of 2 m
◇ Weir closed returns a crest level of 6 m.

With a standard trigger (section 6.4.2) produces the result Figure 6.12. After the head difference exceeds the threshold of 0.3 m, the weir is opened. This has the effect on the the water levels: the water level upstream decreases and the water level upstream increases, so the head difference goes below the threshold, which means that the weir must be closed again. This again increases the head difference towards a value above the threshold, which means that the weir must be opened again.

Figure 6.13 shows a simulation result where a dead band trigger has been applied with a dead band of 0.3 m for b_{up} and 0.05 m for b_{down}. The weir is not opened before the head difference of 0.3 m is reached, but the closing of the weir is not carried out before the head difference decreases to 0.05 m.

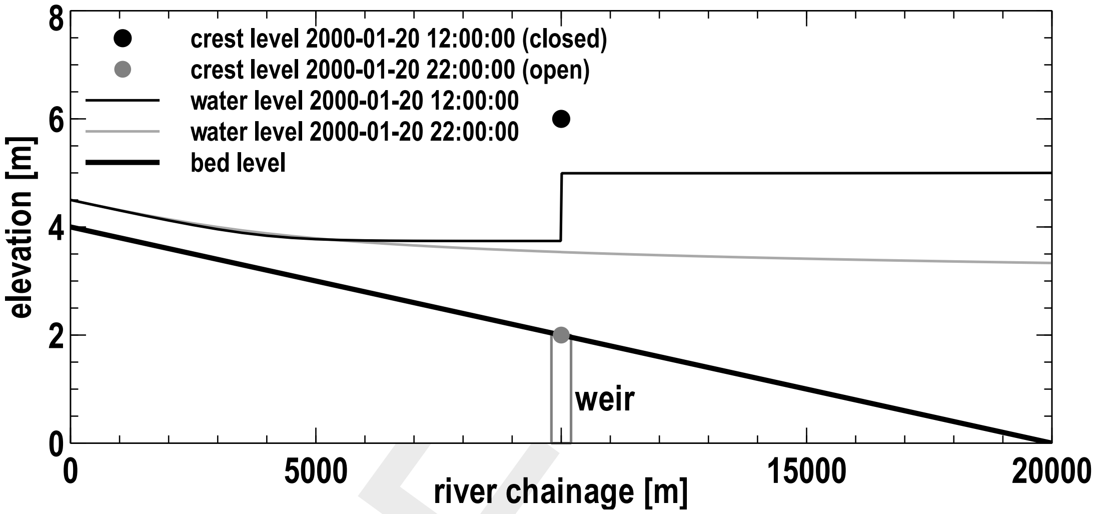


Figure 6.11: *Crest level and water level along chainage for two time stamps. At 12:00 hours the water level on the sea side (right) is higher than on the inland side (left). The weir is closed. At 22:00 hours, the sea is at low tide. The weir is open and water flows towards the sea.*

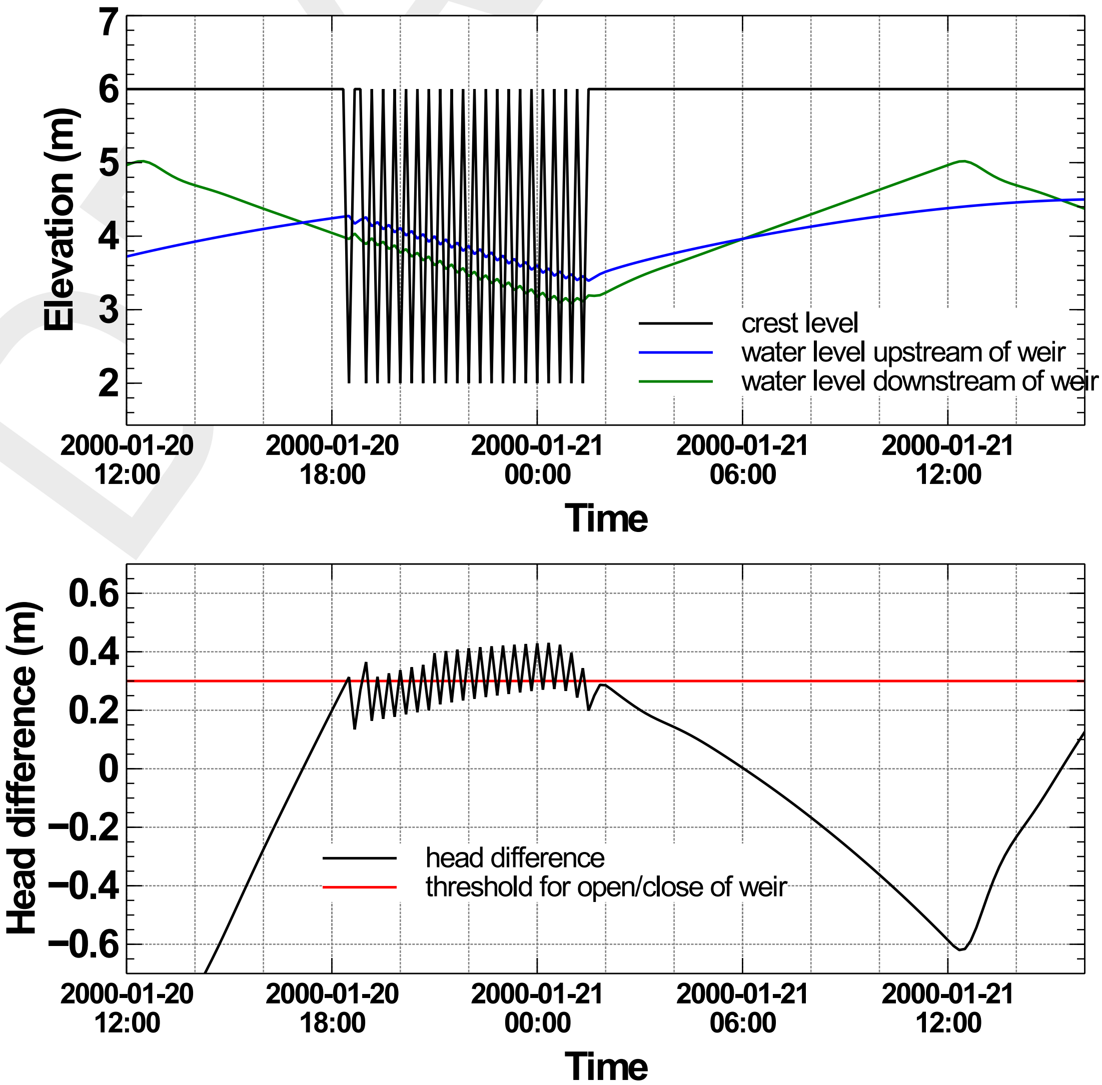


Figure 6.12: *On-off control triggered by head difference*

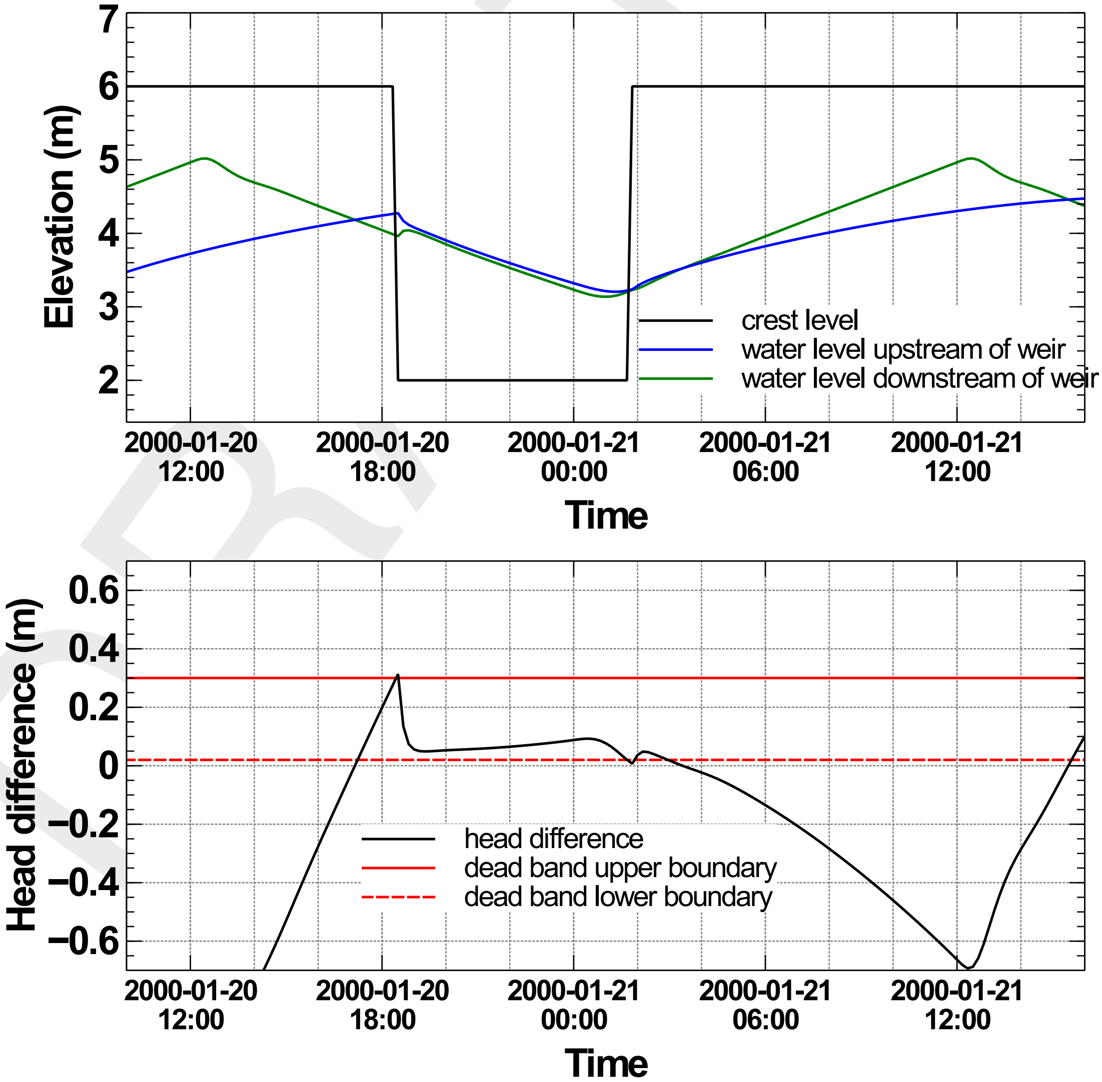


Figure 6.13: *Adding a dead band reduces the number of weir operations*

6.4.5 deadBandTime

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The dead band time trigger checks a time series for a number of subsequent up-crossings n_{up} or down-crossing n_{down} from its current value. If the crossing is observed for at least the user-defined number of time steps, the new value is used. The trigger reads

$$x^k = \begin{cases} x^k & \text{if } \left\{x^{k-n_{\text{up}}+1}, ..., x^k\right\} > x^{k-n_{\text{up}}} \\ x^k & \text{if } \left\{x^{k-n_{\text{down}}+1}, ..., x^k\right\} < x^{k-n_{\text{down}}} \\ x^{k-1} & \text{otherwise} \end{cases} \tag{6.27}$$

An application for this trigger is the activation or deactivation of alarm levels. The increase of alarm level may happen immediately ($n_{\text{up}} = 0$), whereas the decrease of an alarm level could be done only after a number of n time steps have passed ($n_{\text{down}} = n$) without further threshold crossings.

6.4.6 polygonLookup

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The polygon lookup trigger checks, if a point is inside of a set of polygons. If this is the case, it returns the user-defined value of the specific polygon. The point is defined by the values of two time series, referred to as the x_1 and x_2 coordinate of the point. The rule reads

$$y^k = \begin{cases} y_1 & \text{if } \left(x_1^{k-1 \vee k}, x_2^{k-1 \vee k}\right) \in P_1 \\ \vdots & \vdots \\ y_n & \text{if } \left(x_1^{k-1 \vee k}, x_2^{k-1 \vee k}\right) \in P_n \\ y_{default} & \text{otherwise} \end{cases} \tag{6.28}$$

where y is the result of the rule and $\{P_1, \ldots, P_n\}$ is a set of polygons.

Figure 6.14 presents an example for the application of the trigger to the definition of warning level for controlling a lake release in Canton Bern, Switzerland.

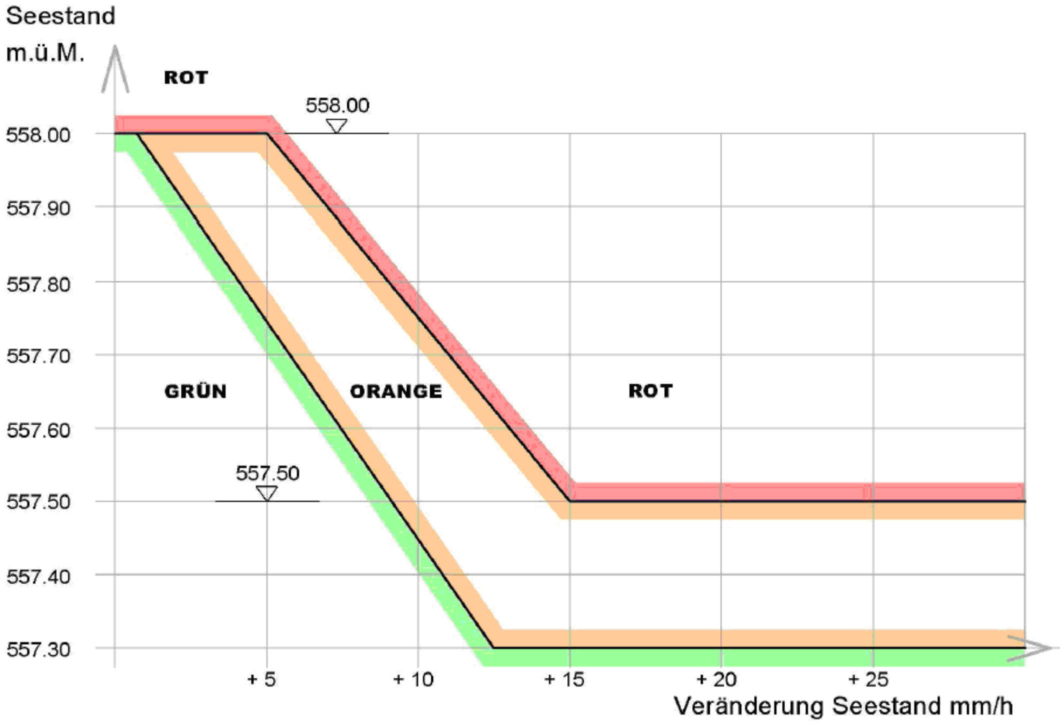


Figure 6.14: *Example for the application of the polygon trigger to the definition of warning levels for controlling a lake release at Lake Thun, Canton Bern, Switzerland*

6.4.7 set

This feature is not yet supported by the D-RTC user interface, but can be configured by modifying the corresponding input file.

The trigger enables a logical combination of other triggers, combined by a logical operator. It reads

$$y^k = x_1^{k-1 \vee k} \wedge x_2^{k-1 \vee k} \tag{6.29}$$

The following operators are supported: $\wedge$ (AND), $\vee$ (OR), $\oplus$ (XOR).

If more than two terms needs to be combined, the set can be used recursively by defining another set as one of the two terms. Therefore, the expression $y^k = x_1^k \wedge (x_2^k \vee x_3^k)$ is represented by a hierarchy of two sets (check the example in the configuration chapter).

6.4.8 Expression

The expression trigger works like the expression component in section 6.2.2. Because triggers are evaluated before components within the time step simulation, it can make sense to use the expression trigger instead of an expression component in order to prepare data of the current time step.

A Data exchange mechanism

In an integrated environment with Deltares Integrated Model Runner (DIMR) or OpenMI (Gregersen *et al.*, 2007), the real-time control model — D-RTC incorporated in GUI framework and RTC-Tools in case of DIMR or OpenMI — is externally coupled with one or more hydro models. According to Morita and Yen (2002) external coupling means data exchange once per time step in both directions (see also Becker and Burzel, 2017), where model output from one model is used by the second model as input, and the output from the second model is fed back to the first model and used as input there.

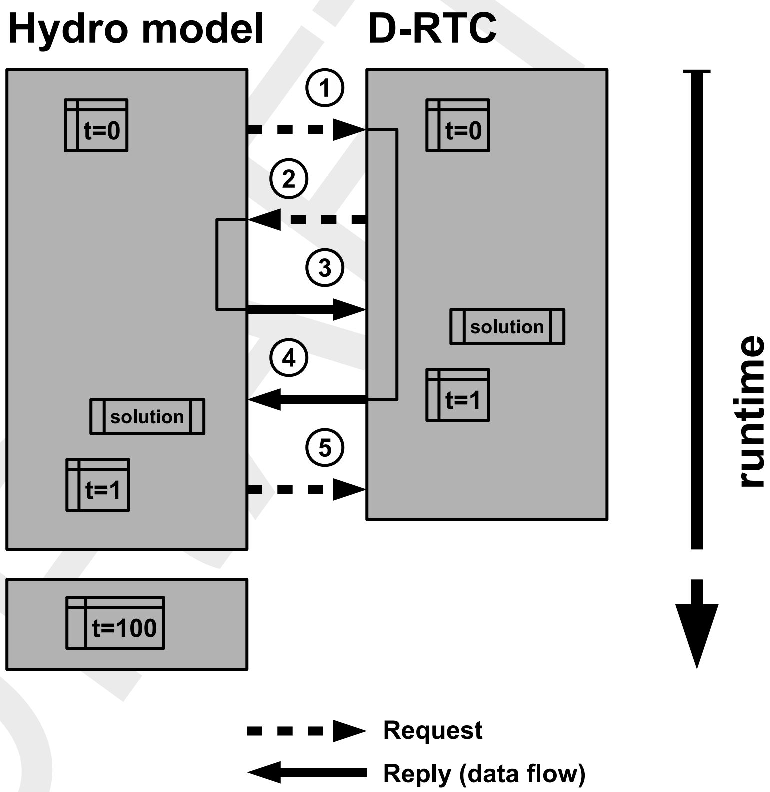


***Figure A.1:** Data exchange within an integrated model composition (modified after Becker and Burzel, 2017, 2016; Becker* et al.*, 2012; Schwanenberg* et al.*, 2011)*

Figure A.1 illustrates the data exchange mechanism for a composition of a hydro model and a D-RTC real-time control model as a request-reply scheme. In order to proceed in simulation time, i.e. to complete the simulation for the unknown time step 1, the hydro model needs values for the controlled parameters from D-RTC and sends a corresponding request to D-RTC (step 1). The D-RTC model, however, needs data from the hydro model to fulfil this request, namely the values of control parameters the decision tree evaluations are based on. D-RTC sends a corresponding data request back to the hydro model (step 2). The hydro model answers with its best guess by sending data from the last computed time step (here: t=0) to D-RTC (step 3). Now D-RTC can carry out its computations for the current time step. Having completed the solution for time step 1, D-RTC is now able to answer the open request for data from the hydro model (step 4). With this information, the hydro model is now able to solve for time step 1. To proceed from time step 1 to time step 2, the procedure is carried out again with a data request for D-RTC (step 5).

The data exchange mechanism results in a time lag between the two models: while the first model gets data from the current time step of the second model, the second model uses data from the previous time step of the first model. For a composition of a hydro model and a feedback control model this is the desired behaviour. The integrated model composition must be arranged accordingly, such that real-time control model computes first and the hydro model computes second for the current time step. DIMR automatically takes care of that. For OpenMI compositions the trigger must be connected to the hydro model (see Becker and Burzel, 2017, for more details).

B D-RTC build instructions on Linux

B.1 Prerequisites

Before building D-RTC, make sure the following applications are present in you environment:

- Compilers: gcc and g++
- cmake 3.0.0 or higher
- tortoise svn

B.2 Build D-RTC

1. Make a directory <D-RTC>, and inside there make other three directories: boost, repo, build.
2. Build boost (see section B.3).
3. Download the following repository inside the repo folder using svn (it might take up to 30 min, due to the inclusion in the third party directory of precompiled boost binaries for windows).
   https://svn.oss.deltares.nl/repos/rtc-tools/branches/FBC-Tools/FBCTools
4. In the build directory run the following cmake command:
   `cmake -DBOOST_ROOT:PATHNAME=/u/carniato/D-RTC/boost/boost_1_65_0_install ../repo/FBCTools`
   Explanation: the first option points to your boost installation directory, the second option points to the source dir of D-RTC (substitute the part "/u/carniato" with the appropriate path in your system).
5. Type make for building D-RTC. Binaries are made inside repo/FBCTools/bin/Release.

B.3 Build boost

1. Download <boost_1_65_0.tar.gz> from
   https://dl.bintray.com/boostorg/release/1.65.0/source/
   and copy it inside the boost directory
2. Untar the tarball in the boost directory:
   `tar -zxvf boost_1_65_0.tar.gz`
3. Create a new directory <boost_1_65_0_install>
4. Move inside boost_1_65_0 (the source code directory) and build boost (do not worry about possible errors in building boost/python, it is not used by D-RTC).
   ```
   ./bootstrap.sh
   ./b2 install -prefix=../boost_1_65_0_install
   ```

B.4 Using D-RTC library with a Delft3D DIMR installation

In order to use the D-RTC binaries in a dimr-run it is sufficient to prepend the LD_LIBRARY_PATH with the D-RTC library directory before executing the <run_dimr.sh> script (<run_dimr.sh> resides in the Delft3D DIMR installation). Below an example:

```
export libdir=/u/carniato/FBC-Tools/repo/FBCTools/bin/Release
export LD\_LIBRARY\_PATH=$libdir:$LD\_LIBRARY\_PATH
/u/carniato/delft3dOSS/trunk/src/bin/run_dimr.sh -d 8 -m dimr.xml
```

Make sure the library name is correct in the <dimr.xml> configuration file (i.e. `FBCTools_BMI`)

```
<component name="myNameRTC">
   <library>FBCTools_BMI</library>
   <workingDir>rtc</workingDir>
   <inputFile>.</inputFile>
```

```
</component>
```

C Configuration outside the user interface

C.1 Modelling in XML files

RTC-Tools is the computational core of D-RTC, and D-RTC writes a set of input files for RTC-Tools. Not all features of RTC-Tools (see Chapter 6) are yet supported by D-RTC. It can be necessary to modify the RTC-Tools input files in order to access such features for more complex feedback control models. Furthermore, it can make sense to introduce model-wide conditions like “flood conditions yes/no” and connect more rules to this condition, or to use the output of one rule for multiple structures. This is not possible in D-RTC, because D-RTC requires that in a control flow always exactly one active path is, and not more than one paths are active.

The RTC-Tools input files are listed in Table C.1. All configuration files are expected in the same working directory. We highly recommend to validate all XML files against the corresponding XSD schema definitions during the model setup. We suggest to use validating XML editors such as XMLSpy (http://www.altova.com/xml-editor/), XML Notepad (http://xmlnotepad.codeplex.com/) or oXygen (http://www.oxygenxml.com/).

The most efficient way to explore configuration options is to check the XSD schemas in an editor such as XMLSpy. Modified files can no longer be used within GUI framework, the RTC-Tools model must be connected to the hydro model via the Deltares Integrated Model Runner (DIMR) or the model coupling standard OpenMI.

Table C.1: *RTC-Tools configuration files*

File	Content	Use
rtcDataConfig.xml	Time series definitions, interface definitions for file io, in-memory data exchange etc.	required
rtcParameterConfig.xml	Externalized parameters for modification in external applications such as Delft-FEWS	optional
rtcRuntimeConfig.xml	Definition of runtime relevant info: time step size, simulation time, file names if deviating from standard naming convention, run mode (simulation, optimization etc.), logging information etc.	required
rtcToolsConfig.xml	The feedback control model with triggers, controllers and expressions	required

C.2 States

Initial conditions must be specified in the file <state_import.xml> according to the OpenDA <treeVector.xsd> format. Note that the date and the time of the initial state is not required, because the run period of the model is already defined in the runtime settings (see Table C.1).

The <state_export.xml> is generated by RTC-Tools and includes the state of the last time step again in the OpenDA <treeVector.xsd> format and meta information for Delft-FEWS for picking it up in the <statePI.xml> file (described in <pi_state.xsd>).

C.3 Time series

Time series are defined in the file <rtcDataConfig.xml> and here arranged as import and as export time series. As the name might indicate, import time series are input for RTC-Tools, and the export time series are filled by RTC-Tools during the computation and written as output. RTC-Tools knows two types of time series:

- ◇ PI time series or as
- ◇ OpenMI exchange item.

The PI time series is the standard option for user input and output, where the data is known beforehand. The data format is the Delft-FEWS-PI format (<pi_timeseries.xsd>), which complies with Delft-FEWS. A PI time series definition in the file <rtcDataConfig.xml> links to time series data in the time series data file <timeseries_import.xml> (input) or <timeseries_export.xml> (output) by the unique combination of locationId, parameterId and, if present, also qualifyerId and ensembleIndex. For time series data files both versions of the Delft-FEWS-PI format are supported: i) pure XML, ii) a combination of XML and binary files. The latter is more efficient in terms of performance, but not human readable.

The time series definition as OpenMI exchange item has been developed for data exchange in intgrated modelling according to the OpenMI standard (Gregersen *et al.*, 2007). Time series data of an import time series of type OpenMI exchange item is not known beforehand, but computed by an outside entity during an integrated model run. It is fed to RTC-Tools during the computation for each time step. Data from export time series of type OpenMI exchange item can be used by other model components of an integrated model. For more information about OpenMI exchange items see Section C.5.

D-RTC uses these two time series types as follows:

- ◇ OpenMI exchange items are used for data that come from another module, like D-Flow FM or D-Flow 1D:
  - ◻ control parameters like a water level computed for an observation point in a D-Flow FM model.
  - ◻ controlled parameters crest level
- ◇ PI time series are used for user-defined time series:
  - ◻ e. g. a time variant setpoint for a PID controller.

For PI time series, the interpolation between data points can be set to block or linear. Figure C.1 illustrates block and linear interpolation.

Extrapolation can be helpful for time series that are shorter than the simulation period. For extrapolation, beside linear and block extrapolation also PERIODIC can be chosen. When BLOCK or LINEAR is chosen as extrapolation option, the time series is extended by block or linear extrapolation, respectively. The periodic option extends the time series by repetition. The periodic option can be helpful to represent tidal patterns of water levels or seasonal patterns of control parameters. The periodic repetition is carried out based on time step, not on date. Lenght and resolution of a time series with periodic extrapolation must be applied with care, because neither leap years are taken into account, nor the different length of a month.

The following configuration example shows a PI time series definition where the interpolation is set to linear and the extrapolation is set to block.

```
<timeSeries id="TimeSeriesId">
```

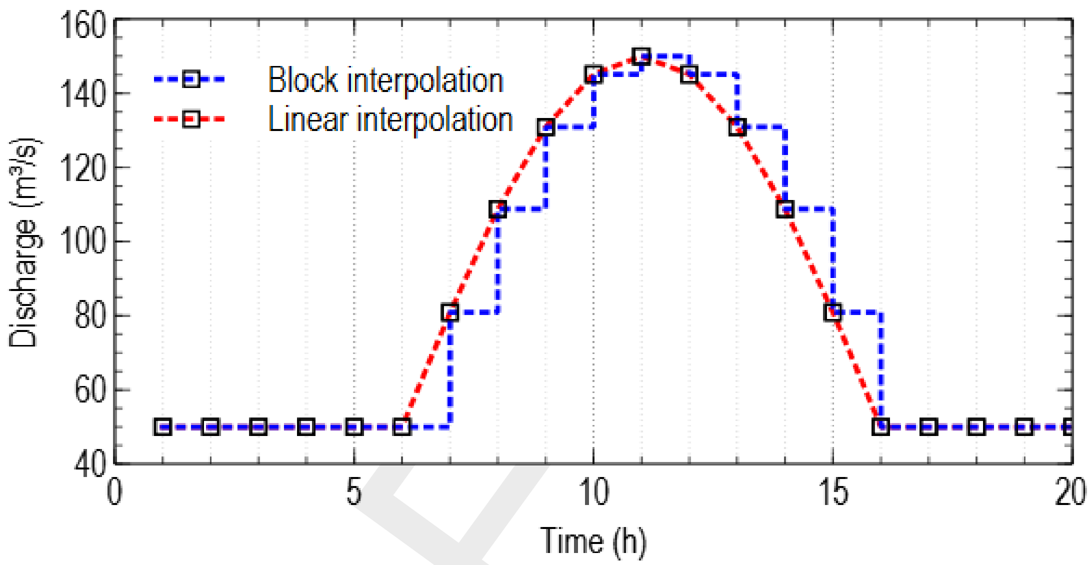


Figure C.1: *Example for block and linear interpolation between data points*

```
        <PITimeSeries>
              <locationId>Location1</locationId>
              <parameterId>Discharge</parameterId>
              <interpolationOption>LINEAR</interpolationOption>
              <extrapolationOption>BLOCK</extrapolationOption>
        </PITimeSeries>
  </timeSeries>
```

C.4 Reference to time series in triggers, rules and expressions

In the file <rtcToolsConfig.xml> triggers, rules and expressions refer to data from time series. The reference to a time series can be explicit and implicit (see also Section 6.1). An explicit reference means that the data from the previous time step is used, while implicit reference means that data assigned to the current time step is used. The default is the explicit reference. The reference option can be configured as follows:

```
<x1Series ref="EXPLICIT">TimeSeriesId</x1Series>
```

or

```
<x1Series ref="IMPLICIT">TimeSeriesId</x1Series>
```

Implicit reference must be used with care: in case the time series is filled during the computation (export time series), data might not yet be present for the current time step.

For integrated models (time series type OpenMI exchange item, see Section C.3), implicit refers to the data that has been provided for the current time step. A time lag on the level of the integrated model might be present though, if this data comes from another model and has been computed for the previous time step by this model.

C.5 RTC-Tools in OpenMI

C.5.1 Introduction

For conjunctive modelling of real-time control with hydraulic processes RTC-Tools is equipped with an interface according to the OpenMI standard. Applications of such conjunctive modelling have been described by Becker *et al.* (2012), Schwanenberg *et al.* (2011) and Becker (2013). Detailed technical information for an OpenMI composition consisting of RTC-Tools and SOBEK is given by Schwanenberg *et al.* (2011) and Becker (2013)

C.5.2 The OMI-file

Below an example of an <*.omi>-file is given. The keyword `Assembly` refers to the location of the dynamic link library with the RTC-Tools computational core. This DLL must provide the OpenMI interface definition. `LinkableComponent` refers to
`Deltares.RtcToolsWrapper.RtcToolsLinkableComponent`,
this is hard-coded in the OpenMI interface definition. Table C.2 explains the meaning of the argument keys.

```
<?xml version="1.0"?>
<LinkableComponent Type="Deltares.RtcToolsWrapper.RtcToolsLinkableComponent"
     Assembly="..\..\RTCToolsOpenMI\bin\Deltares.RtcToolsWrapper.dll"
     xmlns="http://www.openmi.org/LinkableComponent.xsd">
   <Arguments>
      <Argument Key="modelDirectory" ReadOnly="true" Value=".\RtcTools" />
      <Argument Key="MissingValue" ReadOnly="true" Value="0" />
      <Argument Key="OpenMiTimeStepSkip" ReadOnly="true" Value="144" />
      <Argument Key="SchemaLocation" ReadOnly="true"
       Value="..\..\..\RTCTools\xsd\" />
   </Arguments>
</LinkableComponent>
```

Table C.2: *OpenMI argument keys for RTC-Tools models*

argument key	description
`modelDirectory`	Relative or absolute path from the location of the <*.omi>-file to the directory of RTC-Tools input file
`MissingValue`	If this value is provided to RTC-Tools via the `GetValues` method, RTC-Tools will treat it as not existent
`OpenMiTimeStepSkip`	1 means data exchange at every RTC-Tools time step, 2 means data exchange every second time step, and so on. The default value is 1.
`SchemaLocation`	The relative path from the location of the RTC-Tools model directory to the directory with <xsd>-schema definition files. If this argument key is not given, RTC-Tools looks for the xsd files in the RTC-Tools work directory, where the xml files are located.

C.5.3 OpenMI exchange items

OpenMI exchange items are interpreted as an alternative to pre-defined time series, e.g. a Delft-FEWS-PI-time series in <timeseries_import.xml>. Consequently, OpenMI input exchange items and OpenMI output exchange items are defined in the file <rtcDataConfig.xml>.

D Errors and unexpected results

D.1 Time series from expression components used in triggers give unexpected results

Error description

A standard trigger condition evaluates a time series provided by an expression model component (**??**). The trigger reacts unexpected.

Reason

Triggers are always evaluated first in the RTC-Tools programme procedure. The result of the expression component time series is not available for the trigger.

Possible solutions

- Use a trigger of type expression (section 6.4.8).
- Define the input time series explicitely: <x1Series ref="EXPLICIT">. The trigger will use the expression result from the previous time step.

D.2 Values from an import time series do not appear in the result file

Error description

Values of a time series from the <timeseries_import.xml> do not appear in the result file <timeseries_0000.csv>. The column header is present, but the column is empty.

Possible reasons

- The time series is not referenced to in the file <rtcDataConfig.xml> under <importSeries>
- The begin and end time properties <startDate> and <endDate> in the time series header (<header>) do not match the simulation time properties given in <rtcRuntimeConfig.xml>.
- No extrapolation and interpolation options are given in <rtcDataConfig.xml>.
- A combination of <locationId> and <parameterId> is used twice in <rtcDataConfig.xml>.
- The simulation runs under OpenMI and the attribute TimeSeries xsi:schemaLocation directs to a location that is not valid, for example ..\..\..\badFolder\pi_timeseries.xsd.

Solution

- Check the list above and adjust the configuration accordingly.
- When using OpenMI, change the schema location path (attribute TimeSeries xsi:schemaLocation) in such a way that it points towards a valid location, by preference to the location of the binaries ..\..\..\bin\pi_timeseries.xsd.

Error description

Some output variables are missing in the result file <timeseries_0000.csv>, although they are listed in the file <rtcDataConfig.xml> under <exportSeries>.

Possible reasons

- The export time series is defined twice, as import and as export time series.

Possible solutions

The limited memory option is switched on in <rtcRuntimeConfig.xml>. Set it to `false`:

```
<mode>
  <simulation>
    <limitedMemory>false</limitedMemory>
  </simulation>
</mode>
```

Note that the recent versions of RTC-Tools no longer have a limited memory option.

D.3 Index not found in time series model

Error message

The file <diag.xml> says:

```
"int main(int argc, char *argv[]) - error -
int timeSeriesBasics::getScalarIndex(string s) -
index not found in time series model:  "
```

No name is given for the time series RTC-Tools can not find.

Reason

RTC-Tools looks for a time series with an empty name which it cannot find in <rtcDataConfig.xml>.

Possible solution

Look for incomplete items where RTC-Tools writes output for in the file <rtcToolsConfig.xml> and specify an output time series. Make sure that the time series names appear in the file <rtcDataConfig.xml> under <importSeries>. Do not specify a time series without name in <rtcDataConfig.xml>.

Examples for incomplete items are given below:

Example 1: the time series where the status of a standard trigger is to be written is empty:

```
<output>
  <status></status>
</output>
```

Example 2: the target time series of an expression is not specified:

```
<y></y>
```

Example 3: an input time series is not specified:

```
<x1Series ref="IMPLICIT"/>
```

D.4 Instance document parsing failed

Error message

The file <diag.xml> says:

```
error - instance document parsing failed"
level="1"
```

Reason

A file is missing.

Possible solution

Check if all files are available. Most likely the file `state_import.xml` is missing.

D.5 An export time series contains unexpected values

Error

The values of a time series are computed, but they are not correct according to the control flow.

Possible reasons and solution

- The time series is referred to by more than one rule.
- The time series is referred to by more than one rule. One or more rules are not referenced by a trigger. In this case, the rule is always active and overwrites the output of another rule. Delete the rules that are not referred to by a trigger, or integrate it in a trigger to achieve the desired behaviour.

References

Åström, K. J. and T. Hägglund, 1995. *PID controllers: theory, design and tuning*. Instrument Society of America, Research Triangle Park, NC, USA, 2nd ed.

Becker, B., 2013. *Inzet RTC-Tools voor het boezemmodel "Wetterskip Fryslân"*. Report 1205773-000, Deltares, Delft. In Dutch.

Becker, B. and A. Burzel, 2016. *Model coupling with OpenMI / Introduction, basic concepts and live demonstration*. Lecture notes for the webinar held on April 21st, 2016, Deltares, Delft.

Becker, B. and A. Burzel, 2017. "Model Coupling with OpenMI - Introduction of Basic Concepts." In R. Setola, V. Rosato, E. Kyriakides and E. Rome, eds., *Managing the Complexity of Critical Infrastructures / A Modelling and Simulation Approach*, vol. 90 of *Studies in Systems, Decision and Control*, pages 279–299. Springer Open.

Becker, B., C. Sprengers and T. Ochterbeck, 2011. *AMICE: Action 6. Hydraulic Sobek-model for the Dutch part of the Meuse*. Deltares memo 1201007-000-ZWS-0014, Deltares.

Becker, B. P. J., D. Schwanenberg, T. Schruff and M. Hatz, 2012. "Conjunctive real-time control and hydrodynamic modelling in application to Rhine River." In *Proceedings of 10th International Conference on Hydroinformatics*. TuTech Verlag TuTech Innovation GmbH, Hamburg, Germany.

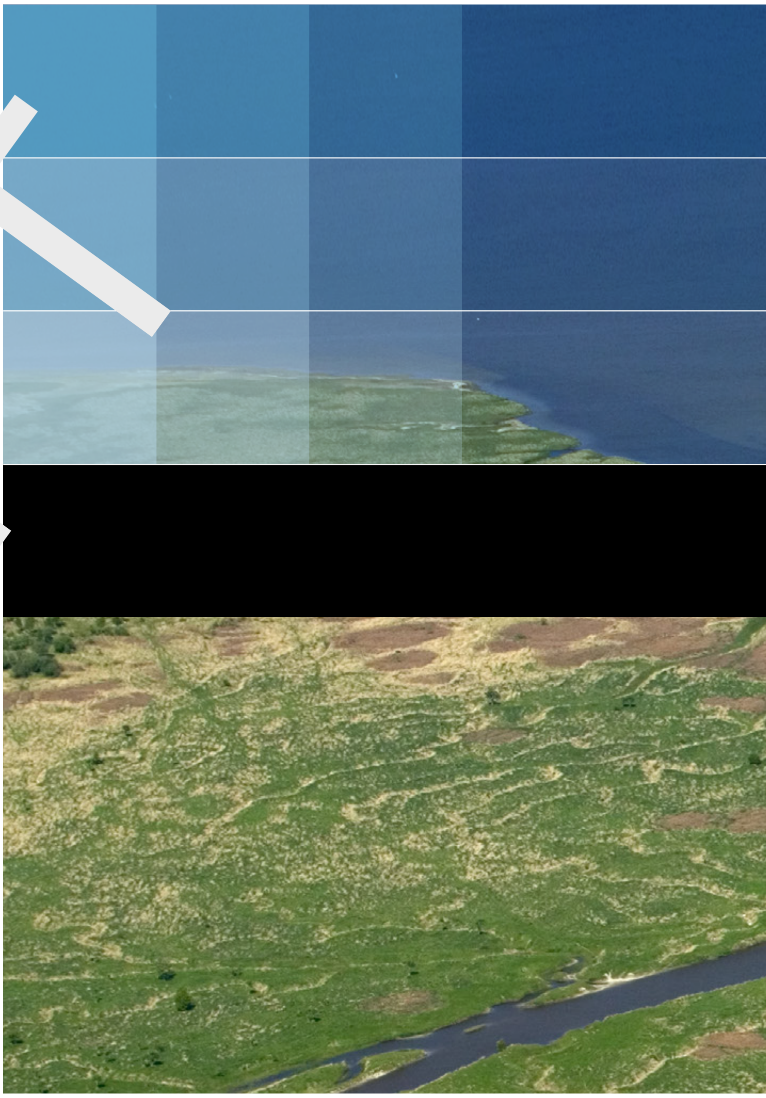
Deltares, 2024. *D-Flow FM Hydro- and Morphodynamics User Manual*. Deltares, 1.1.124 ed.

Gregersen, J., P. Gijsbers and S. Westen, 2007. "OpenMI: Open modelling Interface." *Journal of Hydroinformatics* 9 (3): 175–191. DOI: 10.2166/hydro.2007.023.

Morita, M. and B. Yen, 2002. "Modeling of Conjunctive Two-Dimensional Surface-Three-Dimensional Subsurface Flows." *Journal of Hydraulic Engineering* 128 (2): 184–200. DOI: 10.1061/(ASCE)0733-9429(2002)128:2(184).

Schwanenberg, D., B. Becker and T. Schruff, 2011. *SOBEK-Grobmodell des staugeregelten Oberrheins*. Report no. 1201242-000-ZWS-0014, Deltares. In German.

SOBEK 3 User Manual, 2014. *SOBEK 3 / D-Flow 1D and D-Real Time Control User Manual*. Deltares, Delft. Version: 3.2.1.31793.



Deltares systems

PO Box 177
2600 MH Delft
Boussinesqweg 1
2629 HV Delft
The Netherlands

+31 (0)88 335 81 88
software@deltares.nl
www.deltares.nl/software